



CMS INTEGRATION GUIDE

Integrating a CMS System with the BrightSign Platform

TABLE OF CONTENTS

Introduction	1
Simple File Networking Process	1
Local Storage Structure	2
Pool Directory Structure.....	3
Autoplay File.....	4
Presentation Metadata	4
Zone Parameters	10
Playlist Data.....	14
States	16
Transitions	22
Commands	24
Current Sync (Setup)	32
Current Sync (Runtime).....	39



INTRODUCTION

This guide outlines how to create a customized system for distributing and updating content on the BrightSign platform. The method for content distribution outlined in this guide is known as Simple File Networking: The player is initially configured with an SD card containing setup files; it will then retrieve content, presentation, and scheduling files from a web folder setup. Many resources for deploying this system are provided by BrightSign, while others will need to be generated by the integrator.

Simple File Networking Process

The following is an overview of how the Simple File Networking process works:

1. A number of setup files are placed on an SD card. This includes a *current-sync.xml* file that determines (along with the general networking behavior of the player) the URL that the player should use to download presentation files.
2. The card is inserted into the player, and the player is powered on.
3. Once the player attains a valid Internet connection, it will send an HTTP GET request to the specified URL, attempting to retrieve a runtime version of the *current-sync.xml*, which contains a manifest of all files that need to be on the SD card for runtime operation (content files, *autorun.brs*, *autoplay.xml*, etc.).
4. The player sends HTTP GET requests to the web server for all files specified in the runtime *autoplay.xml* that are not already on the SD card.

Note: *The player uses SHA-1 hash values to determine if a file on the SD card is identical to a file on the web server. This means that the SHA-1 hash values of all files must be calculated before they are published.*

5. The player attempts to run the presentation that was downloaded using the *current-sync.xml* document.
6. The player will periodically compare the SHA-1 hash values of the *current-sync.xml* file on the SD card and the web server. The frequency of these checks is determined by the `<timeBetweenNetConnects>` parameter in the *current-sync.xml* document.
 - a. If the SHA-1 hash values are the same, the presentation will continue uninterrupted.

- b. If the SHA-1 hash values are different, the player will download the new *current-sync.xml* file from the web server. The player will download the necessary files in the manifest as a background process. It will then reboot and play the new presentation.

Local Storage Structure

The SD card image has different file contents depending on whether it is being used for *setup* or *runtime* operation.

Setup

The root of the SD card must include the following when configuring the player:

- [current-sync.xml](#): **Setup file generated by the integrator**
- Brightsign-dumps: Empty folder
- Pool: Empty folder
- Autorun.brs: File provided by BrightSign
- setupCommon.brs: File provided by BrightSign
- setupNetworkDiagnostics.brs: File provided by BrightSign

Runtime

Once the setup process is completed and files have been downloaded from the specified URL, the SD card must include the following:

- brightsign-dumps: Empty folder
- autoplugins.brs: File provided by BrightSign
- autorun.brs: File provided by BrightSign
- [current-sync.xml](#): **File manifest generated by the integrator**
- pool: The folder containing all downloaded files in a sharded directory structure. This folder contains all content files (images, video, etc.), as well as the following files. Note that, when files are placed in the pool folder, they must be named using their SHA-1 hash value (e.g. "sha1-c153d933e1c25734e2f2dd5c1abd5ad106b70f4b").
 - autorun.brs: File provided by BrightSign
 - [autoplay.xml](#): **Presentation file generated by the integrator**

- `autoschedule.xml`: File generated by the application
- `resources.txt`: File provided by BrightSign
- `autopugins.brs`: File generated by application and/or provided by BrightSign
- `_deviceWebPage.html`: Default file provided by BrightSign
- `_deviceIdWebPage.html`: Default file provided by BrightSign

Pool Directory Structure

The pool folder contains all files downloaded from the network. It is organized as follows:

- There are two levels of subfolders contained within the pool. All files are stored in the second level subfolders.
- The location of each file is determined using the SHA-1 hash value of the file. These values are used to identify and match files both in the pool folder and on the web server; for this reason, SHA-1 hash values should be generated for files before they are placed on the web server for synchronization with the player.
- The *second to last* character in the hash value determines the first level of subfolder the file is in, while the *last* character determines the second level of subfolder. For example, a file with the following hash value would be located in `SD:\pool\0\8` because the last two characters are 0 and 8:

```
<hash method="SHA1">b101584aeae4159f8b0643e0d7977c6dd8ffb08</hash>
```

AUTOPLAY FILE

The autoplay file contains all information related to a presentation. It is named using the format `autoplay-[presentation_name].xml`. Like other downloaded files, it is placed into a pool directory based on its hash value. The required parameters of this file are listed below. Note that some parameters have example values that can be customized, while others contain **fixed values** that should never be changed.

Presentation Metadata

`<BrightAuthor version = 6 BrightAuthorVersion="3.8.0.30" type="publish">`: The version and type parameters of this tag are **fixed values**, while the `BrightAuthorVersion` should correspond to the version listed in the corresponding runtime `autorun.brs` file.

- `<meta>`: The beginning of the metadata section
 - `<name>New Presentation</>`: The name of the presentation represented by the autoplay file
 - `<model>XD1230</>`: The model designation of the BrightSign player
 - `<videoMode>1920x1080x60p</>`: The video mode for the presentation. See [this FAQ](#) for a list of available video modes by model.
 - `<monitorOrientation>Landscape</>`: The orientation of the display area. This value can be either Landscape or Portrait.
 - `<videoConnector>HDMI</>`: The video connector used to display the presentation. This value can be HDMI, VGA, or Composite
 - `<deviceWebPageDisplay>Standard</>`: **Fixed value**
 - `<alphabetizeVariableNames>True</>`: A Boolean value that determines if the list of variable names returned by WebKit JavaScript calls should be alphabetized or not
 - `<delayScheduleChangeUntilMediaEndEvent>True</>`: A Boolean value that determines if a presentation change, which is determined by the `autoschedule.xml` file, will only occur after the current media item completes playback (`True`), or if the presentation change will occur immediately (`False`).

- `<htmlEnableJavascriptConsole>False</>`: A Boolean value that determines if the JavaScript console for WebKit should be enabled or disabled
- `<backgroundScreenColor a="255" r="192" g="192" b="192"</>`: A set of values that determine the ARGB color space of the presentation background
- `<monitorOverscan>noOverscan</>`: The overscan setting of presentation. The following parameters can be used:
 - `noOverscan`: Displays the presentation to the edge of the video signal.
 - `overscanActionSafe`: Restricts the presentation display to a screen area that is compatible with most displays.
 - `overscanTitleSafe`: Restricts the presentation display to a screen area that is substantially smaller than the video signal.
- `<gpio0>input</>`: A value that determines whether the GPIO 0 port is configured as an input or output
- `<gpio1>input</>`: A value that determines whether the GPIO 1 port is configured as an input or output
- `<gpio2>input</>`: A value that determines whether the GPIO 2 port is configured as an input or output
- `<gpio3>input</>`: A value that determines whether the GPIO 3 port is configured as an input or output
- `<gpio4>input</>`: A value that determines whether the GPIO 4 port is configured as an input or output
- `<gpio5>input</>`: A value that determines whether the GPIO 5 port is configured as an input or output
- `<gpio6>input</>`: A value that determines whether the GPIO 6 port is configured as an input or output
- `<gpio7>input</>`: A value that determines whether the GPIO 7 port is configured as an input or output
- `<BP900AConfigureAutomatically>True</>`: **Fixed value**
- `<BP900BConfigureAutomatically>True</>`: **Fixed value**
- `<BP900CConfigureAutomatically>True</>`: **Fixed value**
- `<BP200AConfigureAutomatically>True</>`: **Fixed value**
- `<BP200BConfigureAutomatically>True</>`: **Fixed value**
- `<BP200CConfigureAutomatically>True</>`: **Fixed value**
- `<BP900AConfiguration>0</>`: **Fixed value**
- `<BP900BConfiguration>0</>`: **Fixed value**
- `<BP900CConfiguration>0</>`: **Fixed value**

- `<BP200AConfiguration>0</>`: **Fixed value**
- `<BP200BConfiguration>0</>`: **Fixed value**
- `<BP200CConfiguration>0</>`: **Fixed value**
- `<SerialPortConfiguration>`: A serial port configuration entry. The autoplay file should contain six entries listed as `<SerialPortConfiguration>`, one for each serial port. The `<port>` parameter is used to differentiate each entry.
 - `<port>0</>`: The serial port number, which can range from 0 to 5
 - `<baudRate>115200</>`: The baud rate
 - `<dataBits>8</>`: The number of data bits
 - `<parity>N</>`: A value that determines whether parity should be used (Y) or not (N)
 - `<protocol>ASCII</>`: A value that determines whether the serial output should be formatted as ASCII or Binary
 - `<sendEol>CR</>`: A value that determines whether the send EOL is CR or CR+LF
 - `<receiveEol>CR</>`: A value that determines whether the receive EOL is CR or CR+LF
 - `<invertSignals>False</>`: **Fixed value**
 - `<connectedDevice>None</>`: **Fixed value**
- `</SerialPortConfiguration>`: The end of a serial port configuration entry
- `<udpDestinationAddress>255.255.255.255</>`: The destination address for UDP messages sent from the application
- `<udpDestinationPort>5000</>`: The destination port for UDP messages sent from the application
- `<flipCoordinates>false</>`: A Boolean value that indicates whether the x and y coordinates should be reversed for touch screens
- `<touchCursorDisplayMode>auto</>`: The display mode of the cursor when a zone is being used to accept touch events (which are also used to accept USB mouse input):
 - `hide`: Always hides the cursor.
 - `display`: Always displays the cursor.
 - `auto`: Only displays the cursor if there are multiple touch events/regions assigned to the current state.
- `<language>English</>`: **Fixed value**

- `<languageKey>eng</languageKey>`: The date/time localization settings of the application. See the *roResourceManager* entry in the Object Reference Manual for more information.
- `<audio1MinVolume>0</>`: **Fixed value**
- `<audio1MaxVolume>100</>`: **Fixed value**
- `<audio2MinVolume>0</>`: **Fixed value**
- `<audio2MaxVolume>100</>`: **Fixed value**
- `<audio3MinVolume>0</>`: **Fixed value**
- `<audio3MaxVolume>100</>`: **Fixed value**
- `<usbMinVolume>0</>`: **Fixed value**
- `<usbMaxVolume>100</>`: **Fixed value**
- `<hdmiMinVolume>0</>`: **Fixed value**
- `<hdmiMaxVolume>100</>`: **Fixed value**
- `<spdifMinVolume>0</>`: **Fixed value**
- `<spdifMaxVolume>100</>`: **Fixed value**
- `<inactivityTimeout>False</>`: **Fixed value**
- `<inactivityTime>30</>`: **Fixed value**
- `<autoCreateMediaCounterVariables>False</>`: **Fixed value**
- `<resetVariablesOnPresenationStart>False</>`: A Boolean value indicating whether or not User Variables should be reset to their default values when a presentation starts.
- `<networkedVariablesUpdateInterval>300</>`: **Fixed value**
- `<userVariables>`: The User Variables definition section. See **Chapter 11 > Variables** in the BrightAuthor User Guide for more information about parameters associated with User Variables.
 - `<userVariable>`: A section for defining a User Variable
 - `<name>Variable 1</>`: The name of the variable
 - `<defaultValue>42</>`: The default value of the variable
 - `<networked>False</>`: **Fixed value**

- `<liveDataFeedName>Feed 1</>`: The Data Feed associated with the User Variable. This value is only applicable for variables that are *networked* (i.e. their value can be specified using entries in a Live Data feed).
- `<systemVariable>serialNumber</>`: The system variable associated with the User Variable value. This parameter is only applicable to *system* variables. It can have the following values:
 - o `serialNumber`
 - o `ipAddressWired`
 - o `ipAddressWireless`
 - o `firmwareVersion`
 - o `scriptVersion`
 - o `rfChannelCount`
 - o `rfChannelName`
 - o `rfVirtualChannel`
 - o `tunerScanPercentageComplete`
 - o `edidMonitorSerialNumber`
 - o `edidYearOfManufacture`
 - o `edidMonitorName`
 - o `edidManufacturer`
 - o `edidUnspecifiedText`
 - o `edidSerialNumber`
 - o `edidManufacturerProductCode`
 - o `edidWeekOfManufacture`
 - o `activePresentation`
- `<access>Private</ >`: The privacy setting of the User Variable. A `Shared` variable can be accessed by other presentations on the SD card, while a `Private` variable cannot be accessed by other presentations on the SD card. If two presentations use a variable with the same name, the value of the variable with the `Shared` attribute will be used for the variable with the `Private` attribute.

- `</userVariable>`: The end of a User Variable definition
- `</userVariables>`: The end of the User Variables definition section
- `<liveDataFeeds>`: The beginning of the Data Feeds definition section. See **Chapter 11 > Data Feeds** in the BrightAuthor User Guide for more information about parameters associated with Data Feeds.
 - `<liveDataFeed>`: The beginning of a Data Feed definition
 - `<Name>Feed 1</>`: The name of the Data Feed
 - `<url>`
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>brightsign.biz/feed1</>`: The URL where the RSS file associated the Data Feed can be downloaded
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `</url>`
 - `<pluginFilePath></>`: **Fixed value**
 - `<uvPluginFilePath></>`: **Fixed value**
 - `<downloadContent>False</>`: **Fixed value**
 - `<updateInterval>60</>`: A value that determines how often (in seconds) the RSS file should be downloaded
 - `<autoGenerateUserVariables>False</>`: **Fixed value**
 - `<userVariableAccess>Private</>`: **Fixed value**
 - `<uvParserFunctionName></>`: **Fixed value**
 - `</liveDataFeed>`: The end of a Data Feed definition
- `</liveDataFeed>`: The end of the Data Feeds definition section
- `<scriptPlugins>`: The beginning of the script plugins definition section
 - `<scriptPlugin>`: The beginning of a script plugin definition
 - `<name>Custom</>`: The name of the plugin
 - `<path></>`: **Fixed value**

- `</scriptPlugin>`: The end of a script plugin definition
 - `</scriptPlugins>`: The end of the script plugins definition section
 - `<htmlSites>`: The beginning of the HTML sites definition section
 - `<localHTMLSite>`: The beginning of a definition for an HTML site that is stored locally
 - `<name>Site 1</>`: The name of the HTML site
 - `<prefix>Site 1-</>`: The same value as the site `<name>`, but with a hyphen added to the end
 - `<filepath>site.html</>`: The name of the *.html* file associated with the site
 - `</localHTMLSite>`: The end of a definition for a local HTML site
 - `<remoteHTMLSite>`: The beginning of a definition for an HTML site that is accessed remotely
 - `<name>Site 2</>`: The name of the HTML site
 - `<url>`
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>http://brightsign.biz</>`: The URL of the remote HTML site
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `</url>`
 - `</remoteHTMLSite>`: The end of a definition for a remote HTML site
 - `</htmlSites>`:
 - `<presentationIdentifiers></>`: **Fixed value**
 - `<DirectoryLocations></>`: **Fixed value**
- `</meta>`: The end of the meta definition section

Zone Parameters

The `<zones>` section contains information about each zone in the presentation:

- `<zones>`: The beginning of the zones definition section
 - `<zone>`: The beginning of a zone definition.

- `<name>Zone 1</>`: The user-defined name for the zone
- `<id>1</>`: An integer that functions as a unique identifier for the zone
- `<x>0</>`: The horizontal coordinates for the top-left corner of zone rectangle
- `<y>0</>`: The vertical coordinates for the top-left corner of the zone rectangle
- `<width>1920</>`: The width of the zone rectangle
- `<height>1080</>`: The height of the zone rectangle
- `<type>VideoOrImages</>`: The zone type. The following are valid zone types, which may be limited by the BrightSign player model:
 - **Images**: Displays image files.
 - **VideoOrImages**: Displays video, image, or audio files.
 - **VideoOnly**: Displays video or audio files.
 - **AudioOnly**: Plays audio files.
 - **EnhancedAudio**: Plays audio in a cross-faded playlist. For more details, see **Chapter 10 > Editing Audio Only and Enhanced Audio zones** in the BrightAuthor User Manual.
 - **BackgroundImage**: Places an image in the background of the presentation. All other zones appear on top of the background image.
 - **Ticker**: Displays RSS feeds or pre-defined text strings.
 - **Clock**: Displays the time or date.
- `<zoomValue>1</>`: **Fixed value**
- `<horizontalOffset>0</>`: **Fixed value**
- `<verticalOffset>0</>`: **Fixed value**
- `<zoneSpecificParameters>`: Settings that are related to the audio/video output of a particular zone. The applicable zone types are listed next to each parameter below.
 - `<zOrderFront>True</>`:(Video or Images, Video Only) A Boolean value that determines the z-ordering of the zone. Setting this parameter to `True` places the zone in front of a second Video Only zone or Video or Images zone, while setting this parameter to `False` places the zone behind a second Video Only zone or Video or Images zone. This parameter has no effect if there is only one Video Only and/or Video or Images zone.

- `<viewMode>Scale to Fill</>`:(Video or Images, Video Only) The view mode of video displayed within the zone:
 - `Scale to Fill`: Scales the video to fill the zone without maintaining the aspect ratio.
 - `Letterboxed and Centered`: Centers the video and adds black borders on the top and bottom. The aspect ratio is maintained.
 - `Fill Screen and Centered`: Centers the video and fills the screen. The aspect ratio is maintained.
- `<imageMode>Scale to Fit</>`:(Video or Images, Images) The view mode of images displayed in the zone:
 - `Center Image`: Centers the image without scaling. This may result in cropping if the image is too large.
 - `Scale to Fit`: Scales the image to fit the zone. The image is displayed as large as possible while keeping the correct aspect ratio.
 - `Scale to Fill`: Scales the image to fill the zone without maintaining the aspect ratio.
 - `Scale to Fill and Crop`: Scales the image to completely fill the zone while maintaining the aspect ratio.
- `<audioOutput>Analog Audio</>`:(Video or Images, Video Only, Audio Only) **Fixed value**
- `<audioMode>Multichannel Surround</>`:(Video or Images, Video Only, Audio Only) **Fixed value**
- `<audioMapping>Audio-1</>`:(Video or Images, Video Only, Audio Only) **Fixed value**
- `<analogOutput>PCM</>`:(Video or Images, Video Only, Audio Only) The analog audio output mode. Setting this parameter to `PCM` enables analog audio output from the zone, while setting this parameter to `None` disables analog audio output from the zone.
- `<analog2Output>None</>`:(Video or Images, Video Only, Audio Only) **Fixed value**
- `<analog3Output>None</>`:(Video or Images, Video Only, Audio Only) **Fixed value**
- `<hdmiOutput>None</>`:(Video or Images, Video Only, Audio Only) The HDMI audio output mode. Setting this parameter to `PCM` enables HDMI audio output from the zone, while setting this parameter to `None` disables HDMI audio output from the zone.

- `<spdifOutput>PassThrough</>`: (Video or Images, Video Only, Audio Only) The SPDIF audio output mode. Setting this parameter to `PassThrough` enables SPDIF audio output from the zone, while setting this parameter to `None` disables SPDIF audio output from the zone.
- `<usbOutput>None</>`: (Video or Images, Video Only, Audio Only) **Fixed value**
- `<audioMixMode>Stereo</>`: (Video or Images, Video Only, Audio Only) The audio mixing mode. This parameter accepts the following values: `Stereo`, `Left`, `Right`.
- `<videoVolume>100</>`: (Video or Images, Video Only) The initial volume (0-100) for audio output from video files
- `<audioVolume>100</>`: (Video or Images, Video Only, Audio Only) The initial volume (0-100) for audio output from audio files
- `<minimumVolume>0</>`: (Video or Images, Video Only, Audio Only) Sets the minimum volume for the zone (0-100).
- `<maximumVolume>100</>`: (Video or Images, Video Only, Audio Only) Sets the maximum volume for the zone (0-100). Note that the **Maximum Volume** setting will be overridden by other volume settings in a presentation, including the **Initial Volume** setting for the zone and volume commands attached to events or states.
- `<liveVideoInput>S-Video</>`: (Video or Images, Video Only) **Fixed value**
- `<liveVideoStandard>NTSC-M</>`: (Video or Images, Video Only) **Fixed value**
- `<brightness>128</>` (Video or Images, Video Only) **Fixed value**
- `<contrast>64</>`: (Video or Images, Video Only) **Fixed value**
- `<saturation>64</>`: (Video or Images, Video Only) **Fixed value**
- `<hue>0</hue>`: (Video or Images, Video Only) **Fixed value**
- `<textWidget>`: (Ticker) The beginning of the text widget definition section
 - `<numberOfLines>3</>`: The number of text lines displayed at a time in the zone
 - `<delay>5</>`: The amount of time (in seconds) each line displays within the zone
 - `<rotation>0</>`: The rotation of the text (in degrees). This parameter can accept one of the following integer values: 0, 90, 180, 270.

- `<alignment>left</code>: The alignment of the text. This parameter can accept one of the following integer values: left, center, right.`
 - `<scrollingMethod>0</code>: The text animation setting. Setting this parameter to 0 specifies an animated ticker, while setting it to 1 specifies static text.`
- `</textWidget>:(Ticker)` The end of the text widget definition section
- `<widget>:(Ticker, Clock)` The beginning of the section that defines the widget appearance
 - `<foregroundColor a="255" r="255" g="255" b="255" />`: A set of values that determine the ARGB color space of the text
 - `<backgroundTextColor a="255" r="255" g="255" b="255" />`: A set of values that determine the ARGB color space of the text backdrop
 - `System</code>: The text font. The System value indicates the default character set, but any True Type Font (e.g. wingdings.ttf) can be specified for this parameter.`
 - `<fontSize>0</code>: Fixed value`
- `</widget>: (Ticker, Clock)` The end of the section that defines the widget appearance
- `<displayTime>False</code>: (Clock) A Boolean value that determines whether the Clock zone should display the time (True) or the date (False).`

Playlist Data

The content of the `<playlist>` section is different depending on the zone `<type>`. The playlist parameters for each zone `<type>` are outlined below.

Images, Video or Images, Video Only, Audio Only, and Enhanced Audio Zones

- `<playlist>`
 - `<name>Playlist 1</name>`: The user-defined name for the playlist
 - `<type>interactive</type>`: **Fixed Value**
 - `<states>`: The section for defining all images, videos, audio tracks, and other multimedia items in a zone playlist. This section also includes definitions for all transitions between states.
 - `<initialState>Image 1.jpg</code>: The name of the first <state> in the presentation`

- `<state></>`: A state definition. See the [States](#) section for a definition of each state type.
- `<transition>`: A transition definition. See the [Transitions](#) section for a definition of each transition type.
- `</states>`: The end of the states definition section

Background

- `<playlist>`
 - `<name>Playlist 2</name>`: The user-defined name for the playlist
 - `<type>non-interactive</type>`: **Fixed value**
 - `<backgroundImageItem>`
 - `<file name="Image 1.jpg" path =""/>`: The file name of the image for the Background zone. A Background zone can contain only a single state (i.e. an image file).
 - `</backgroundImageItem>`
 - `<states>`
 - `<initial State />`: **Fixed value**
 - `</states>`
- `</playlist>`

Ticker

- `<playlist>`
 - `<name>Playlist 3</>`: The user-defined name for the playlist
 - `<type>non-interactive</>`: **Fixed value**
 - `<rssDataFeedPlaylistItem>`
 - `<liveDataFeedName>Feed 1</>`: The `<Name>` of a Data Feed that has been defined in the [liveDataFeeds](#) section of the play file.
 - `</rssDataFeedPlaylistItem>`
 - `<states>`
 - `<initial State />`: **Fixed value**
 - `</states>`

- `</playlist>`

Clock

- `<playlist>`
 - `<name>Playlist 4</>`: The user-defined name for the playlist
 - `<type>non-interactive</>`: **Fixed value**
 - `<clockItem />`: **Fixed value**
 - `<states>`
 - `<initial State />`: **Fixed value**
 - `</states>`
- `</playlist>`

States

States represent individual media items within a playlist. The following zone types use a playlists: Images, Video or Images, Video Only, Audio Only, Enhanced Audio. The different state types are listed below:

Image

Contains an image file.

- `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value at the beginning of the `<states>` section, this state will begin the presentation.
 - `<imageItem>`: The beginning of an image item definition
 - `<file name="Slide 1.jpg"/>`: The name of the image file
 - `<fileIsLocal>True</>`: **Fixed value**
 - `<slideDelayInterval>3</>`: The amount of time (in seconds) the image will be displayed on screen
 - `<slideTransition>No effect</>`: The transition effect that will occur at the end of the display interval. See this [FAQ](#) for a list of all available transition effects.

- `</imageItem>`: The end of an image item definition
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).
 - `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. This section can contain multiple `<brightSignCmd>` entries (one for each command).
 - `<brightSignCmd></>`
- o `</state>`

Video

Contains a video file.

- o `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value at the beginning of the `<states>` section, this state will begin the presentation.
 - `<videoItem>`: The beginning of an image item definition
 - `<file name= "Video 1.mov" />`: The name of the video file
 - `<fileIsLocal>True</>`: **Fixed value**
 - `<videoDisplayMode>2D</videoDisplayMode>`: The 3D display mode of the video file. This parameter can be assigned either 2D or 3D values.
 - `</videoItem>`: The end of the video item definition
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).
 - `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. Unlike entry commands, the `<brightSignCmd>` definitions are contained within the `<brightSignExitCommands>` section.
 - `<brightSignCmd></>`
- o `</state>`

Audio

Contains an audio file.

- o `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value at the beginning of the `<states>` section, this state will begin the presentation.
 - `<audioItem>`: The beginning of an audio item definition
 - `<file name= "Track 1.mp3" />`: The name of the audio file
 - `<fileIsLocal>True</>`: **Fixed value**
 - `</audioItem>`: The end of the audio item definition
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).
 - `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. Unlike entry commands, the `<brightSignCmd>` definitions are contained within the `<brightSignExitCommands>` section.
 - `<brightSignCmd></>`
- o `</state>`

HTML5

Contains an HTML5 page.

- o `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value specified above, this state will begin the presentation.
 - `<html5Item>`: The beginning of the HTML state definition
 - `<name>New HTML Page</>`: The user-assigned name of the state
 - `<htmlSiteName> Local Page</>`: The `<name>` of an HTML site specified in the [<htmlSites>](#) section of the metadata
 - `<enableExternalData>True</>`: A Boolean value that should be set to `True` if the HTML page uses assets from multiple locations

- `<enableMouseEvents>True</>`: A Boolean value that indicates whether mouse and touch events should be enabled for the HTML page (via HTML or JavaScript)
 - `<displayCursor>True</>`: A Boolean value that indicates whether or not a cursor should be displayed when a USB mouse is connected to the player
 - `<timeOnScreen>0</>`: **Fixed value**
 - `</html5Item>`: The end of the HTML state definition
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).
 - `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. Unlike entry commands, the `<brightSignCmd>` definitions are contained within the `<brightSignExitCommands>` section.
 - `<brightSignCmd></>`
- o `<state>`

Live Video

Displays live video input from the HDMI-In or RF-In connectors (XD1230 only).

- o `<state>`
- `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value specified above, this state will begin the presentation.
 - `<liveVideoItem>`: The beginning of the live video state definition
 - `<volume>100</>`: The volume percentage of the video
 - `<timeOnScreen>0</>`: **Fixed value**
 - `<overscan>True</>`: A Boolean value indicating whether the video size should be matched to an area that is compatible with most displays
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).

- `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. Unlike entry commands, the `<brightSignCmd>` definitions are contained within the `<brightSignExitCommands>` section.
 - `<brightSignCmd></>`
- `</state>`

Media RSS Feed

Plays images, video, or audio from a media RSS (MRSS) feed.

- `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value specified above, this state will begin the presentation.
 - `<rssImageItem>`
 - `<rssSpec title="Daily MRSS Feed" usesBSNDynamicPlaylist="False" />`: The specification for the MRSS feed, which should include the `<title>` of the feed as it is specified in the RSS document. The `usesBSNDynamicPlaylist` parameter is a **fixed value**.
 - `<urlSpec>`
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>http://brightsign.biz/Feeds/MRSS_1</>`: The download URL for the MRSS feed.
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `</urlSpec>`
 - `</rssImageItem>`
 - `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the state is entered. Each state can have multiple `<brightSignCmd>` entries (one for each command).

- `<brightSignExitCommands></>`: The section for defining all [commands](#) that occur when the state is exited. Unlike entry commands, the `<brightSignCmd>` definitions are contained within the `<brightSignExitCommands>` section.
 - `<brightSignCmd></>`
- `</state>`

Media List (Image List, Video List, Audio List)

Allows you to gather files of a particular media type (image, video, or audio) into a single playlist that can also be treated like a state.

- `<state>`
 - `<name>State 1</>`: The name of the state. If this value is the same as the `<initialState>` value specified above, this state will begin the presentation.
 - `<mediaListItem>`
 - `<mediaType>image</>`: The type of media in the Media List. This value can be one of the following: image, video, or audio.
 - `<advanceOnMediaEnd>True</>`: A Boolean value that determines if the Media List should cycle to the next file in the list after an audio or video file ends. If this parameter is set to `False`, the Media List will play one audio or video file each time it is transitioned to, remaining on that file until the playlist transitions away from the Media List state.
 - `<advanceOnImageTimeout>True</>`: A Boolean value that determines if the Media List should cycle to the next file in the list after an image file times out. This parameter is equivalent to the `<advanceOnMediaEnd>` parameter above, except it applies to Image Lists only.
 - `<playFromBeginning>True</>`: A Boolean value that determines if the Media List should start from the beginning each time it is transitioned to. If this parameter is set to `False`, then subsequent transitions to the Media List will cause the next file in the list to be played. This parameter has no affect if the `<advanceOnMediaEnd>/<advanceOnImageTimeout>` value is set to `False`.

- `<imageTimeout>5</>`: The amount of time (in seconds) that an image will display before transitioning to the next item on the list. This parameter is only applicable to Image Lists that have an `<advanceOnImageTimeout>` parameter set to `True`.
- `<shuffle>False</>`: A Boolean value that determines if the order of media items in the Media List should be randomized each time the Media List is transitioned to.
- `<next>`: The section for defining an event that will transition to the next media item in the Media List.
 - `<userEvent></>`: The `<userEvent>` section of a transition (see the Transitions section for more details). The following transitions are supported within Media Lists:
 - BP900A/BP900B/BP900C
 - BP200A/BP200B/BP200C
 - GPIO
 - Serial Input
 - Keyboard Input
 - Zone Message
- `</next>`
- `<previous>`

Transitions

Transitions define how the presentation moves between media states. Each transition type has a trigger and a set of parameters that determine the behavior of the transition. There can be multiple transitions to or from a single media state.

Timeout

This transition triggers after a specified amount of time.

- `<transition>`
 - `<sourceMediaState>State 1</>`: The name of the state that will be exited when the transition event is triggered

- `<userEvent>`
 - `<name>timeout</>`: **Fixed value**
 - `<parameters>`
 - `<parameter>3</>`: The amount of time that will elapse (in seconds) before the transition is triggered
 - `</parameters>`
- `</userEvent>`
- `<targetMediaState>State 2</>`: The name of the state that will be entered after the transition event is triggered.
- `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the transition event is triggered. Each transition can have multiple `<brightSignCmd>` entries (one for each command).
- `<assignInputToUserVariable>False</>`: **Fixed value**
- `<assignWildcardToUserVariable>False</>`: **Fixed value**
- `</transition>`

Media End

This transition triggers once the audio or video file finishes playback.

- `<transition>`
 - `<sourceMediaState>State 1</>`: The name of the state that will be exited when the transition event is triggered
 - `<userEvent>`
 - `<name>mediaEnd</>`: **Fixed value**
 - `<parameters>`
 - `<parameter />`: **Fixed value**
 - `</parameters>`
 - `</userEvent>`
 - `<targetMediaState>State 2</>`: The name of the state that will be entered after the transition event is triggered.

- `<brightSignCmd></>`: A section for defining a [command](#) that occurs when the transition event is triggered. Each transition can have multiple `<brightSignCmd>` entries (one for each command).
- `<assignInputToUserVariable>False</>`: **Fixed value**
- `<assignWildcardToUserVariable>False</>`: **Fixed value**
- `</transition>`

Work in Progress...

Commands

Commands are used to change numerous properties associated with the presentation. They can be assigned to states or transitions.

Set Zone Volume

Specifies the a new volume for all audio/video files in the zone.

- `<BrightsignCmd>`
 - `<name>SetZoneVolume</>`: **Fixed value**
 - `<customUI>True</customUI>`: **Fixed value**
 - `<command>`
 - `<name>setZoneVolume</>`: **Fixed value**
 - `<parameter>`
 - `<name>zoneId</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>1</>`: The [zone <id>](#) specifying which zone the volume change should affect
 - `<parameterValueItemText>`
 - `</parameterValue>`

- o <itemName />: **Fixed value**
 - o <validationRule>-1</>: **Fixed value**
- </parameter>
- <parameter>
 - o <name>volume</>: **Fixed value**
 - o <parameterValue>
 - <parameterValueItemText>
 - <value>90</>: The new volume (0-100) for the zone
 - <parameterValueItemText>
 - o </parameterValue>
 - o <validationRule>8</>: **Fixed value**
- </parameter>
- </command>
- o </brightSignCmd>

SetAudio

Specifies audio output parameters for the zone.

- o <BrightsignCmd>
 - <name>SetAllAudioOutputs</>: **Fixed value**
 - <customUI>True</customUI>: **Fixed value**
 - <command>
 - <name>setAllAudioOutputs</>: **Fixed value**
 - <parameter>
 - o <name>zoneId</>: **Fixed value**
 - o <uiElementName />: **Fixed value**
 - o <parameterValue>
 - <parameterValueItemText>

- `<value>1</>`: The [zone <id>](#) specifying which zone the audio settings change should affect
 - `<parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
 - `</parameter>`
 - `<parameter>`
 - `<name>analog</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>pcm</>`: The analog audio output mode. Setting this parameter to PCM enables analog audio output from the zone, while setting this parameter to None disables analog audio output from the zone.
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
 - `</parameter>`
 - `<parameter>`
 - `<name>analog2</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>none</>`: **Fixed value**
 - `</parameterValueItemText>`
 - `</parameterValue>`

- `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
- `</parameter>`
- `<parameter>`
 - `<name>analog3</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>none</>`: **Fixed value**
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
- `</parameter>`
- `<parameter>`
 - `<name>hdmi</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>pcm</>`: The audio setting of the HDMI output:
 - pcm: Supplies the decoded audio signal through the HDMI connector.
 - passthrough: Supplies the un-decoded audio signal through the HDMI connector. Use this option if the zone audio (for example, AC3 Dolby Digital) is being decoded on an external device.
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**

- `</parameter>`
- `<parameter>`
 - `<name>spdif</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>pcm</>`: The audio setting of the SPDIF output:
 - pcm: Supplies the decoded audio signal through the SPDIF connector.
 - none: Supplies the un-decoded audio signal through the SPDIF connector. Use this option if the zone audio (for example, AC3 Dolby Digital) is being decoded on an external device.
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
- `</parameter>`
- `<parameter>`
 - `<name>usb</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>pcm</>`: **Fixed value**
 - `</parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
- `</parameter>`
- `</command>`

- <command>
 - <name>setAudioMode</name>: **Fixed value**
 - <parameter>
 - <name>zoneId</>: **Fixed value**
 - <uiElementName />: **Fixed value**
 - <parameterValue>
 - <parameterValueItemText>
 - <value>1</>: The [zone <id>](#) specifying which zone the audio mixing change should affect
 - <parameterValueItemText>
 - </parameterValue>
 - <itemName />: **Fixed value**
 - <validationRule>-1</>: **Fixed value**
 - </parameter>
 - <parameter>
 - <name>mode</>: **Fixed value**
 - <uiElementName />: **Fixed value**
 - <parameterValue>
 - <parameterValueItemText>
 - <value>1</>: The audio mixing setting, which can be one of the following values: left, right, or stereo
 - <parameterValueItemText>
 - </parameterValue>
 - <itemName />: **Fixed value**
 - <validationRule>-1</>: **Fixed value**
 - </parameter>
- </command>
- <command>

- <name>configureAudioResources</>
- </command>
- </brightSignCmd>

Change Connector Volume

Changes the volume of the specified connector for all zones in the presentation.

- <BrightsignCmd>
 - <name>IncrementConnectorVolume</>: The type of change that will be applied to the connector volume: The following types can be used:
 - SetConnectorVolume
 - IncrementConnectorVolume
 - DecrementConnectorVolume
 - MuteConnectorVolume
 - UnmuteConnectorVolume
 - <customUI>True</>: **Fixed value**
 - <command>
 - <name>IncrementConnectorVolume</>: The same command type as the <name> parameter above.
 - <parameter>
 - <name>connector</>: **Fixed value**
 - <uiElementName />: **Fixed value**
 - <parameterValue>
 - <parameterValueItemText>
 - <value>Analog</>: The connector that will be affected by the volume change. This parameter accepts the following values: Analog, HDMI, SPDIF
 - <parameterValueItemText>
 - </parameterValue>
 - <itemName />: **Fixed value**

- `<validationRule>-1</>`: **Fixed value**
 - `</parameter>`
 - `<parameter>`
 - `<name>volume</>`: **Fixed value**
 - `<uiElementName />`: **Fixed value**
 - `<parameterValue>`
 - `<parameterValueItemText>`
 - `<value>10</>`: The value applied to the connector volume. This value has different effects depending on the type of change being applied to the connector:
 - Set: Sets the volume (0-100) to the specified value.
 - Increment: Increments the current volume using the specified value.
 - Decrement: Decrements the current volume using the specified value.
 - Mute: Has no effect on the mute action.
 - Unmute: Has no effect on the un-mute action.
 - `<parameterValueItemText>`
 - `</parameterValue>`
 - `<itemName />`: **Fixed value**
 - `<validationRule>-1</>`: **Fixed value**
 - `</parameter>`
- `</brightSignCmd>`

CURRENT SYNC (SETUP)

During the setup phase (before the player has downloaded content from the remote content host), the *current-sync.xml* file contains player configuration information. The required parameters of this file are listed as below. Note that some parameters have example values that can be customized, while others contain **fixed values** that should never be changed.

- `<sync version="1.0" name="Simple Networking">`: **Fixed value**
 - `<meta>`
 - `<client>`
 - `<base>http://www.brightsign.com/files</>`: The base URL of the web server that acts as the remote content host. The run-time `current-sync.xml` file must be located in this directory.
 - `<next>/current-sync.xml</>`: **Fixed value**
 - `<event>EVENT</>`: **Fixed value**
 - `<error>ERROR</>`: **Fixed value**
 - `<deviceerror>DEVICEERROR</>`: **Fixed value**
 - `<uploadusage>UPLOADUSAGE</>`: **Fixed value**
 - `<nowplaying>NOWPLAYING</>`: **Fixed value**
 - `<getfile>GETFILE</>`: **Fixed value**
 - `<uploadlogs></>`: **Fixed value**
 - `<timezone>PST</>`: The time zone that the player will use for synchronization and scheduling operations. See the *roSystemTime* entry in the Object Reference Manual for a list of all available time zones.
 - `<unitName>Test Device</unitName>`: The user-assigned name for the player
 - `<unitNamingMethod>unitNameOnly</unitNamingMethod>`: The naming method for the player:
 - `unitNameOnly`: The player serial number is not appended to the end of the `<unitName>`.

- `appendUnitIDToUnitName`: The player serial number is appended to the end of the `<unitName>`.
- `<unitDescription></>`: A snippet of optional user-created text that can serve as a description for the player (e.g. purpose, intended deployment, location)
- `<timeBetweenNetConnects>300</>`: A value that specifies how often the player should check the *current-sync.xml* file on the web server to determine if a content update is necessary
- `<contentDownloadsRestricted>no</>`: A value that determines if content downloads should be restricted to a certain part of the day. The range start and range length parameters below determine when content downloads should take place.
- `<contentDownloadRangeStart>0</>`: The starting point of the content download window. This value is measured in minutes from midnight: For example, a value of 600 would specify a start time at 10:00AM
- `<contentDownloadRangeLength>0</>`: The length of time the content download window should last. This value is measured in minutes from the range start point: For example, a value of 180, combined with a range start of 600, would configure a content download window lasting from 10:00AM until 1:00PM.
- `<timeBetweenHeartbeats>300</>`: **Fixed value**
- `<heartbeatsRestricted>no</>`: **Fixed value**
- `<specifyHostname>no</>`: A value that determines if the player should use a custom hostname. By default, the player serial number is used to generate a unique hostname.
- `<hostname></>`: The custom hostname. This parameter should only be included in the document if `<specifyHostname>` is set to *yes*.
- `<useProxy>no</>`: A value that determines if the player should connect to the Internet using a proxy server.
- `<proxy></>`: The proxy server address. If a user name and password needs to be provided, use the following format: `<user name>:<password>@<proxy server address>` (e.g. "admin:root@myproxy.com").

- `<useWireless>yes</>`: A value that determines if the player should connect to the network using the Wireless Module add-on.
- `<ssid>Router 1</>`: The SSID of the wireless router that the player should connect to. This value should only be specified if `<>` is set to `yes`.
- `<passphrase></>`: **Fixed value**. The BrightSign platform uses a proprietary algorithm to convert hash values for passwords. This allows passwords to be securely stored in openly accessible `.xml` files on the SD card. Contact BrightSign for more information about this algorithm.
- `<timeServer>http://time.brightsignnetwork.com</>`: The time server that the player will use to retrieve the current time
- `<useDHCP>yes</>`: A value that determines whether the player should connect to the network using DHCP. If this parameter is set to `no`, then static IP settings will need to be specified below.

Note: *If wireless is disabled, the `<useDHCP>` parameter refers to the Ethernet connection; if wireless is enabled, `<useDHCP>` refers to the wireless connection and `<useDHCP_2>` refers to the Ethernet connection.*

- `<staticIPAddress></>`: The static IP address of the player
- `<subnetMask></>`: The subnet mask of the player
- `<gateway></>`: The gateway assigned to the player
- `<dns1</>`: The first DNS entry assigned to the player
- `<dns2</>`: The second DNS entry assigned to the player
- `<dns3</>`: The third DNS entry assigned to the player
- `<rateLimitModeOutsideWindow>unlimited</>`: The bandwidth limit applied to downloads that occur outside the content download window [specified above](#). This parameter can be either `unlimited`, `default`, or `specified` (the rate is specified in the next parameter).
- `<rateLimitRateOutsideWindow>0</>`: A value between 0 and 2000 that represents the content download limit (in Kbit/s) applied outside the content download window
- `<rateLimitModeInWindow>unlimited</>`: The bandwidth limit applied to downloads that occur inside the content download window [specified above](#). This parameter can be either `unlimited`, `default`, or `specified` (the rate is specified in the next parameter).

- `<rateLimitRateInWindow>0</rateLimitRateInWindow>`: A value between 0 and 2000 that represents the content download limit (in Kbit/s) applied inside the content download window
- `<rateLimitModeInitialDownloads>unlimited</>`: The bandwidth limit applied to initial content downloads. This parameter can be either `unlimited`, `default`, or `specified` (the rate is specified in the next parameter).
- `<rateLimitRateInitialDownloads>0</rateLimitRateInitialDownloads>`
- `<useDHCP_2>yes</>`: A value that determines whether the player should connect to the Ethernet network using DHCP. This parameter is only used if both the WiFi and Ethernet connections are configured. If the player is configured to use only one or the other, this parameter (and all other networking parameters that end with `_2` below) should not be included.
- `<staticIPAddress_2></ >`
- `<subnetMask_2></ >`
- `<gateway_2></ >`
- `<dns1_2</>`
- `<dns2_2</>`
- `<dns3_2</>`
- `<rateLimitModeOutsideWindow_2>default</>`
- `<rateLimitRateOutsideWindow_2>0</>`
- `<rateLimitModeInWindow_2>specified</>`
- `<rateLimitRateInWindow_2>1024000</>`
- `<lwsConfig>status</>`: **Fixed value**
- `<lwsUserName>admin</>`: The user name of the local web server
- `<lwsPassword></>`: **Fixed value**. The BrightSign platform uses a proprietary algorithm to convert hash values for passwords. This allows passwords to be securely stored in openly accessible `.xml` files on the SD card. Contact BrightSign for more information about this algorithm.
- `<lwsEnableUpdateNotifications>yes</>`: **Fixed value**
- `<playbackLoggingEnabled>yes</>`: A value that determines whether playback logging should be enabled.

- `<eventLoggingEnabled>yes</>`: A value that determines whether event logging should be enabled
- `<stateLoggingEnabled>yes</>`: A value that determines whether state logging should be enabled
- `<diagnosticLoggingEnabled>yes</>`: A value that determines whether diagnostic logging should be enabled

Note: *Logs are saved on the SD card at regular intervals. They can also be uploaded to the web server in a standard BrightAuthor configuration. This functionality may be extended to the CMS Integration Package in the future. See this [FAQ](#) for more information about the data generated by each log type.*

- `<uploadLogFilesAtBoot>no </uploadLogFilesAtBoot>`: **Fixed value**
- `<uploadLogFilesAtSpecificTime>no</uploadLogFilesAtSpecificTime>`: **Fixed value**
- `<uploadLogFilesTime>0</uploadLogFilesTime>`: **Fixed value**
- `<networkConnectionPriorityWired>0</>`: A value that determines the connection priority of the Ethernet connection (0 or 1)
- `<networkConnectionPriorityWireless>1</>`: A value that determines the connection priority the wireless connection (0 or 1)
- `<contentTypeEnabledWired>True</>`: A Boolean value that determines whether content data can be downloaded over the Ethernet connection
- `<textFeedsDataTypeEnabledWired>True</>`: A Boolean value that determines whether RSS data can be downloaded over the Ethernet connection
- `<mediaFeedsDataTypeEnabledWired>True</>`: A Boolean value that determines whether MRSS data can be downloaded over the Ethernet connection
- `<healthDataTypeEnabledWired>True</>`: A Boolean value that determines whether system heartbeat data can be uploaded over the Ethernet connection
- `<logUploadsDataTypeEnabledWired>True</>`: A Boolean value that determines whether logs can be uploaded over the Ethernet connection

- `<dwsEnabled>no</>`: A value that determines whether the Diagnostic Web Server should be enabled on the player
- `<dwsPassword></>`: **Fixed value**. The BrightSign platform uses a proprietary algorithm to convert hash values for passwords. This allows passwords to be securely stored in openly accessible `.xml` files on the SD card. Contact BrightSign for more information about this algorithm.
- `<idleScreenColor>FFB164FF</idleScreenColor>`: The ARGB value of the screen color that is displayed when the player has no presentation to play
- `<networkDiagnosticsEnabled>False</networkDiagnosticsEnabled>`: A Boolean value that determines whether network diagnostics are enabled. If enabled, network diagnostics display on the screen connected to the player during the startup process.
- `<testEthernetEnabled>True</testEthernetEnabled>`: A Boolean value that determines whether the Ethernet connection should be tested during network diagnostics. If enabled, the player will indicate whether or not it was able to attain an IP address over Ethernet.
- `<testWirelessEnabled>False</testWirelessEnabled>`: A Boolean value that determines whether the wireless connection should be tested during network diagnostics. If enabled, the player will indicate whether or not it was able to attain a an IP address on the wireless network
- `<testInternetEnabled>True</testInternetEnabled>`: A Boolean value that determines whether the Internet connection should be tested during network diagnostics. If enabled, the player will indicate whether or not it was able to attain a connection to remote services on the Internet.
- `</client>`
- `<server>`
 - `<account>ACCOUNT</account>`: **Fixed value**
 - `<user>un</user>`: **Fixed value**
 - `<password>pw</password>`: **Fixed value**
 - `<enableUnsafeAuthentication>False</enableUnsafeAuthentication>`: **Fixed value**
 - `<group>Simple Networking</group>`: **Fixed value**

- `</server>`
 - `</meta>`
 - `<files />`: **Fixed value**
- `</sync>`

CURRENT SYNC (RUNTIME)

At runtime, the *current-sync.xml* acts as a manifest file detailing every presentation and content item necessary to play an updated presentation.

- `<sync version="1.0" name="Simple Networking">`
 - `<meta>`: The beginning of the metadata section
 - `<client>`
 - `<base>http://www.brightsign.com/files</>`: The base URL of the web server that acts as the remote content host
 - `<next>/current-sync.xml</>`: **Fixed value**
 - `<event>EVENT</>`: **Fixed value**
 - `<error>ERROR</>`: **Fixed value**
 - `<deviceerror>DEVICEERROR</>`: **Fixed value**
 - `<uploadusage>UPLOADUSAGE</>`: **Fixed value**
 - `<nowplaying>NOWPLAYING</>`: **Fixed value**
 - `<getfile>GETFILE</>`: **Fixed value**
 - `<uploadlogs></>`: **Fixed value**
 - `<timeBetweenNetConnects>300</>`: A value that specifies how often the player should check the *current-sync.xml* file on the web server to determine if a content update is necessary
 - `<contentDownloadsRestricted>no</>`: A value that determines if content downloads should be restricted to a certain part of the day. The range start and range length parameters below determine when content downloads should take place.
 - `<contentDownloadRangeStart>0</>`: The starting point of the content download window. This value is measured in minutes from midnight: For example, a value of 600 would specify a start time at 10:00AM
 - `<contentDownloadRangeLength>0</>`: The length of time the content download window should last. This value is measured in minutes from the range start point: For example, a value of

180, combined with a range start of 600, would configure a content download window lasting from 10:00AM until 1:00PM.

- `<playbackLoggingEnabled>yes</>`: A value that determines whether playback logging should be enabled.
- `<eventLoggingEnabled>yes</>`: A value that determines whether event logging should be enabled
- `<stateLoggingEnabled>yes</>`: A value that determines whether state logging should be enabled
- `<diagnosticLoggingEnabled>yes</>`: A value that determines whether diagnostic logging should be enabled
- `<uploadLogFilesAtBoot>no </uploadLogFilesAtBoot>`: **Fixed value**
- `<uploadLogFilesAtSpecificTime>no</uploadLogFilesAtSpecificTime>`: **Fixed value**
- `<uploadLogFilesTime>0</uploadLogFilesTime>`: **Fixed value**
- `<enableSerialDebugging>True</>`: **Fixed value**
- `<enableSystemLogDebugging>True</>`: **Fixed value**
- `</client>`
- `<server>`
 - `<account>ACCOUNT</account>`: **Fixed value**
 - `<user>un</user>`: **Fixed value**
 - `<password>pw</password>`: **Fixed value**
 - `<enableUnsafeAuthentication>False</enableUnsafeAuthentication>`: **Fixed value**
 - `<group>Simple Networking</group>`: **Fixed value**
- `</server>`
- `</meta>`: The end of the metadata section
- `<files>`: The beginning of the files section, which contains details of all files that should be stored on the SD card. If the synchronization process works correctly, the player will download all of the files listed in this section that are not currently on the SD card.

- `<download>`: The beginning of the definition for a file that should be synchronized
 - `<name>Image 1.jpg</>`: The file name. If the file is referenced in the *autoplay.xml* file, then the file names must match exactly between the *current-sync.xml* file and the *autoplay.xml* file.
 - `<hash method="SHA1">df2a31626923d3fd7fa79a35c38b9bb14ef5ed4c</>`: The hash of the file contents using the SHA-1 algorithm
 - `<size>14684</>`: The size of the file (in bytes)
 - `<headers inherit="no />`: **Fixed value**
 - `<chargeable>no</chargeable>`: **Fixed value**
- `</download>`: The end of the file definition
- `<delete>`
 - `<pattern>*.brs</pattern>`: **Fixed value**
- `</delete>`
- `<delete>`
 - `<pattern>*.rok</pattern>`: **Fixed value**
- `</delete>`
- `<delete>`
 - `<pattern>*.bsfw</pattern>`: **Fixed value**
- `</delete>`
- `<ignore>`
 - `<pattern>*</pattern>`: **Fixed value**
- `</ignore>`
 - `</files>`: The end of the files definition section
- `</sync>`