DC20 Digital Cinema Server

Automation Guide

Version 1.1

Effective Date: September 2009
Document # 9301H362001.1

Datasat Digital Entertainment
9631 Topanga Canyon Place
Chatsworth, CA 91311
USA

Notices

This product contains software proprietary to Datasat Digital Entertainment and Real Image Media Technologies Pvt. Ltd. and are protected by US and International copyright law. Unauthorized reproduction or disclosure, in whole or in part, is strictly prohibited.

WARRANTY/LIMITATION OF LIABILITY

EXCEPT FOR THE LIMITED WARRANTY PROVIDED HEREIN, THIS PRODUCT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND. DATASAT DIGITAL ENTERTAINMENT EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS AND IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DATASAT DIGITAL ENTERTAINMENT DOES NOT WARRANT THAT THIS PRODUCT WILL MEET USER'S REQUIREMENTS, OR THAT THE OPERATION OF THIS PRODUCT WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT DEFECTS IN THIS PRODUCT WILL BE CORRECTED. UNDER NO CIRCUMSTANCES, INCLUDING NEGLIGENCE, SHALL DATASAT DIGITAL ENTERTAINMENT, OR ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS, BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE) ARISING OUT OF THE USE, MISUSE OR INABILITY TO USE THIS PRODUCT OR RELATED DOCUMENTATION.

Copyright

DC20 Digital Cinema Server Installation and Operating Guide, Version 1.1

The content of this publication is subject to change without notice. Datasat Digital Entertainment assumes no obligation to notify you of any changes or updates. While Datasat Digital Entertainment, Inc. believes this publication is accurate, due to ongoing improvements and revisions, Datasat Digital Entertainment cannot guarantee the accuracy of printed material, nor can it accept responsibility for errors or omissions. Datasat Digital Entertainment may periodically publish updates and revisions to this publication as it deems necessary.

DTS and the DTS logo are registered trademarks of DTS, Inc.

Qube and the Qube logo are registered trademarks of Real Image Media Technologies Pvt. Ltd.

All other trademarks are the properties of their respective owners.

Document No. 9301H362001.1

September 2009

Record of Changes

Manual Version / Date	Description
1.0 / May 2009	Pre-production release
1.1/ September 2009	Production release

Table of Contents

Limited Warranty

Datasat Digital Entertainment software is warranted against defects for a period of 90 days from the date of the original purchase. DC20 hardware is warranted against defects in material and workmanship for a period of one year from the date of the original purchase. This warranty includes parts and labor. THIS WARRANTY EXCLUDES UNITS THAT HAVE BEEN MODIFIED, REPAIRED BY UNAUTHORIZED PERSONNEL, OR DAMAGED BY ABUSE, IMPROPER OPERATION, DIRT, MOISTURE OR STATIC ELECTRICITY. PROPER OPERATION OF THIS EQUIPMENT REQUIRES THE USE OF A SURGE PROTECTOR. SUPPLYING POWER TO THE DC20 UNIT WITHOUT THE USE OF A SURGE PROTECTOR WILL VOID THIS WARRANTY.

Products being returned under warranty should be sent to Datasat Digital Entertainment freight prepaid in the original or equivalent packaging. Please contact Datasat Digital Entertainment prior to shipping any parts or merchandise to receive a Return Merchandise Authorization number. Datasat Digital Entertainment's shipping address is:

Datasat Digital Entertainment
Attn: RMA #
9631 Topanga Canyon Place
Chatsworth, CA 91311
U.S.A.

Contact Information

To contact Datasat Digital Entertainment's Technical Support staff:

- - toll free telephone within USA and Canada (888)-428-2268
- - telephone worldwide +1-818-401-4253
- - fax +1-866-448-6802
- - email techsupport@datasatdigital.com

Product Improvements and Upgrades

Datasat Digital Entertainment reserves the right to make changes and/or improvements to its products, without notification and without incurring any obligation to incorporate such changes or improvements in products previously sold or shipped.

To receive notification about upgrades or bulletins that may become available from time to time, please complete the enclosed Warranty Card and mail or fax it to Datasat Digital Entertainment.

Software License Agreement

IMPORTANT: READ THIS SOFTWARE LICENSE ("LICENSE") CAREFULLY BEFORE USING THE DATASAT DIGITAL ENTERTAINMENT SOFTWARE. THE SOFTWARE RESIDING ON THE DC20 DIGITAL CINEMA SERVER AND ANY ACCOMPANYING MATERIALS (COLLECTIVELY, THE "SOFTWARE") ARE LICENSED, NOT SOLD. BY USING THE DC20 DIGITAL CINEMA SERVER, YOU ARE ACCEPTING AND AGREEING TO THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, DO NOT USE THE DC20 DIGITAL CINEMA SERVER.

1. Grant. Subject to your compliance with the terms of this License, Datasat Digital Entertainment hereby grants you a non-exclusive, non-transferable, revocable, limited license to use the Software

solely for your internal business purposes and solely in connection with the DC20 Digital Cinema Server. All rights not expressly granted to you are reserved by Datasat Digital Entertainment.

2. Ownership. All right, title and interest in and to the Software, all physical copies thereof and all intellectual property embodied therein are and shall remain vested in Datasat Digital Entertainment. You do not own the Software, physical copies thereof or any intellectual property embodied therein.
3. Restrictions. You may not (i) copy, distribute, sublicense, lease, transfer or use for external purposes the Software; (ii) reverse engineer, decompile, disassemble, translate, modify, attempt to reveal the source code of the Software, or (iii) create derivative works from the Software.
4. Termination. Datasat Digital Entertainment may terminate this license at any time if you violate the terms of this license. Upon termination, you must destroy the Software and related documentation.
5. LIMITATION ON WARRANTY. YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT USE OF DATASAT DIGITAL ENTERTAINMENT SOFTWARE AND HARDWARE IS AT YOUR SOLE RISK. DATASAT DIGITAL ENTERTAINMENT DOES NOT WARRANT THAT THE OPERATION OF THE SOFTWARE OR THE HARDWARE WILL BE UNINTERRUPTED OR ERROR FREE. EXCEPT FOR THE LIMITED WARRANTY HEREIN AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE DATASAT DIGITAL ENTERTAINMENT SOFTWARE AND HARDWARE PROVIDED “AS IS,” WITH ALL FAULTS AND WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSES, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS, ALL OF WHICH ARE HEREBY EXPRESSLY DISCLAIMED.
6. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES, INCLUDING, WITHOUT LIMITATION, NEGLIGENCE, SHALL DATASAT DIGITAL ENTERTAINMENT OR ITS DIRECTORS, OFFICERS, EMPLOYEES OR AGENTS BE LIABLE FOR PERSONAL INJURY, OR ANY INCIDENTAL, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, LOSS OF DATA, BUSINESS INTERRUPTION OR ANY OTHER COMMERCIAL DAMAGES OR LOSSES, ARISING OUT OF OR RELATED TO THE USE OF THE DATASAT DIGITAL ENTERTAINMENT SOFTWARE OR HARDWARE, HOWEVER CAUSED, REGARDLESS OF THE THEORY OF LIABILITY AND EVEN IF DATASAT DIGITAL ENTERTAINMENT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL DATASAT DIGITAL ENTERTAINMENT’S LIABILITY EXCEED THE AMOUNT PAID BY YOU TO DATASAT DIGITAL ENTERTAINMENT. SOME STATES DO NOT ALLOW THE LIMITATION AND/OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.
7. General. This License shall be governed by the laws of the State of California as applied to contracts performed wholly therein. In any action brought by either party arising out of this License, the prevailing party shall be entitled to recover its reasonable attorney’s fees and costs. If any provision of this License is held by a court of competent jurisdiction to be unenforceable or if a provision of this License is waived by either party, the remainder of this License shall remain in full force and effect. You may not assign this License without the prior, express written consent of Datasat Digital Entertainment.

1. DC20 Automation Basics

The DC20 digital cinema server uses Automation *Cues* that are attached to the *Start* or the *End* of various *Compositions* of a *Show.* These Cues can be *Offset* by a positive time duration within the Composition when attached to the Start of a Composition (i.e. occur at a preset time after the Start of the Composition) or Offset by a negative time duration within the Composition when attached to the End of a Composition (i.e. occur at a preset time before the End of the Composition).

These Automation Cues are managed on the DC20 system's Web UI and automate various events in a Show, executing them precisely at the desired time on externally attached devices such as the digital projector, the cinema audio processor, a lighting control system, etc.

The DC20 system provides several pre-defined Cues and allows the addition of any number of custom Cues as well. Each Cue, whether pre-defined or custom, consists of one or more *Actions,* each Action being executed on one *Device.* Multiple *Instances* of a Device can also be setup on the DC20 system (for example, two identical lighting controllers, both attached to the DC20 but performing independently).

This section discusses the setup of various Devices including standard Devices such as the GPIO and DLP Cinema™ Projectors from Christie, Barco and NEC. It also looks at sample Devices in other categories such as Lighting Controllers.

Automation support can currently be implemented on the DC20 using Ethernet,GPIO (General Purpose Input/Output) or RS-232C Serial control. The specific devices to be loaded on the DC20 system and the interfacing parameters of the device are setup by modifying the file *"AutomationDevices.xml"* in the path *"C:\Program Files\Real Image\Qube 2\"* (henceforth referred to as the *Qube Program Folder*). Standard files for commonly used devices such as the DLP Cinema Projectors are also available in the same path.

The various Automation Cues setup on a DC20 system are available on the Cues tab of the Shows page on the Web User Interface. The *"AutomationCues.xml"* file, also in the Qube Program Folder contains the actual Automation Cue names and their definitions. The "AutomationDevices.xml" file contains information about the Device, the port used for communication, and also the credentials needed for accessing the target device.

The Actions that can be executed on each target device are defined as an xml or as a binary *Dynamic Link Library (DLL)* with a *Method Definition* for each Action. As defined, an Automation Cue is thus a sequence of one or more Actions, executed on one or more attached Devices.

2. Customization of “AutomationDevices.xml” and “AutomationCues.xml”

Each Device to be controlled by the DC20 server has to be configured for the server to recognize and control its functions. This configuration can be done by editing the “AutomationDevices.xml” file in the Qube Program Folder. One or more automation Devices can be added to the “AutomationDevices.xml” file along with the specific parameters required for that Device. A user-friendly name can also be given to each Instance of a Device.

The types of Devices that are currently available are:

a. **ASCII Devices** – These are Devices that communicate using simple text or numeric commands and are configured using an xml definition file. These Devices work over an IP address and Port or over a Serial Communications link.

b. **Binary Devices** – These are pre-programmed Devices that communicate over more complex protocols and are made available as Dynamic Link Libraries (or dll’s). These Devices also work over an IP address and Port or over a Serial Communications link.

c. **Qube Device** – This is for internal control of the DC20 Server by the automation system and may be required, for example, to Pause the DC20 Server while an external command completes execution and then to resume Play.

3. Device and Configuration Values in “AutomationDevices.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<Devices xmlns="http://schemas.qubecinema.com/Automation/Devices/2008-01-26">
        <Device name="Device Name" class="Device Class" enable="true">
                <Configuration>
                            <Key name="File" value="file name of the device"/>
                            <Key name="Address" value="address value"/>
                            <Key name="Port" value="port value"/>
                    </Configuration>
                    <Init>
                            <Parameter name="UserName" value="User Name" />
                            <Parameter name="Password" value="Password"/>
                    </Init>
            </Device>
    </Devices>
```

Shown above is a sample “AutomationDevices.xml” file. The values to be used for the customization of this file are as follows:

3.1. Device Values

- **Device Name:** The user-friendly Device Name. This name must be unique. Multiple versions or *Instances* of a Device can exist but must each have a unique Device Name.
- **Device Class:** This value is dependent on the type of device:
    - Binary: Fully qualified Class name in the binary file
    - ASCII Control over IP: *Qube.Automation.StreamDevice.TCP*
    - ASCII Control over Serial link: *Qube.Automation.StreamDevice.Serial*
    - Qube Device: *Qube.Automation.SMS, Dalapathi*
- **Enable:** *true* or *false* to enable or disable this Device

3.2. Configuration Values

- **File:** The file name of the device
- **Address:** For IP control, this is the IP Address of the device and for Serial control, this provides the parameters for the serial communications in the format "COM_Port, Baud_Rate, Parity, Data_Bits, Stop_Bits" separated by commas
- **Port:** For IP control, this is the IP Port of the device and for Serial control, this Key must not be present

3.3. Init Values

- **UserName:** Where credentials are necessary to control the external device, the User Name parameter is provided here
- **Password:** Where credentials are necessary to control the external device, the Password parameter is provided here. Please note that the User Name and Password parameters need to be used as necessary by the actual Automation Device and are provided here only to be passed on to the Automation Device as parameters.

3.4. Structure of "AutomationCues.xml"

"AutomationCues.xml" is a file that defines the various Cues in a DC20 system. As previously mentioned, a Cue is a list of one or more Actions to be executed on Device Instances. Currently there are only two types of commands that are supported in "AutomationCues.xml" – *InvokeMethod* and *Sleep.*

- **InvokeMethod:** Invokes an Action available on a particular Device Instance. The required *Parameters* for InvokeMethod are the following:
    - **Name:** Name of an Action available on the Device
    - **Device:** Name of the Instance of the Device as defined in "AutomationDevices.xml"

Where the Action requires one or more custom Parameters, these must also be provided for each InvokeMethod definition in the "AutomationCues.xml" file.

The following is an example without any custom Parameters:

```
<InvokeMethod name="Pause" device="ThisQubeServer"/>
```

The following is an example with custom Parameters:

```
<InvokeMethod name="SwitchChannel" device="CP2000ZX">
        <Parameter name="Channel" value="102"/>
</InvokeMethod>
```

- **Sleep:** Waits for the specified duration. Essentially, this command does nothing for the specified time period in order to provide for desired gaps between multiple automation events. There is only one required Parameter for the Sleep command:

  Duration: Duration in a whole number of seconds

The following is an example of the Sleep command:

```
<Sleep duration="3" />
```

4. Automation of the DC20 Device

The DC20 Server can itself be controlled by the automation if necessary. This is practically required in order to Pause the server while an external Action executes and to resume Play when the external Action has completed.

- **Play:** Plays the Show if the DC20 server is in Pause mode. Does not require any custom Parameters.
- **Pause:** Pauses the playback of the current Show if the DC20 server is in Play mode. Does not require any custom Parameters.
- **Toggle:** Toggles the current Show between Play and Pause modes. Does not require any custom Parameters.

The following sample shows how the DC20 Device can be configured in the "AutomationDevices.xml" file with a friendly name of "Me".

```
<?xml version="1.0" encoding="utf-8"?>
<Devices xmlns="http://schemas.qubecinema.com/Automation/Devices/2008-01-26">
        <Device name="Me" class="Qube.Automation.SMS, Dalapathi" enable="true"/>
</Devices>
```

5. Automation of DLP Cinema™ Projectors

Once the projector has been configured in the AutomationDevices.xml file, the file for the particular projector in use has to be configured. The standard files for DLP Cinema™ projectors are available in the Qube Program Folder. They are:

a. Barco.dll for all Barco DLP Cinema projectors

b. Christie.xml for all Christie DLP Cinema projectors

c. NEC.dll for all NEC DLP Cinema projectors

Actions supported by each projector's Automation Device

Below is the List of Actions supported by the supplied Automation files for each projector brand:

5.1. Barco DLP Cinema Projectors

- **ExecuteMacro(MacroName):** MacroName is the parameter which specifies the name of the macro file on the Barco projector that will be executed
- **ShutterOpen:** Opens the projector's mechanical shutter or dowser
- **ShutterClose:** Closes the projector's mechanical shutter or dowser
- **LampOn:** Switches on the projector's lamp
- **LampOff:** Switches off the projector's lamp

5.2. Christie DLP Cinema Projectors

- **SwitchChannel(Channel):** Channel is the custom Parameter which specifies the channel number to be selected on the Christie projector. The value specified must be preceded by a "1" and must be a two digit number padded with a leading zero if necessary. For example, to select Channel "3", the Channel value to be specified is "103"
- **ShutterOpen:** Opens the projector's mechanical shutter or dowser
- **ShutterClose:** Closes the projector's mechanical shutter or dowser
- **Init(UserName, Password):** The Christie projector requires the external device to login using preset credentials before it will execute any commands
- **LampOn:** Switches on the projector's lamp
- **LampOff:** Switches off the projector's lamp

5.3. NEC DLP Cinema Projectors

- **SwitchInput(MacroKey):** MacroKey is a parameter that specifies the input number to be switched to
- **ShutterOpen:** Opens the projector's mechanical shutter or dowser
- **ShutterClose:** Closes the projector's mechanical shutter or dowser
- **LampOn:** Switches on the projector's lamp
- **LampOff:** Switches off the projector's lamp

Sample Christie ASCII Automation Device Definition file

The following is an example of the Christie projector automation definition xml file:

```
<?xml version="1.0" encoding="utf-8" ?>
<StreamDevice name="Christie"
xmlns="http://schemas.qubecinema.com/Automation/StreamDevice/2008-01-26">
        <Init>
                <Parameter name="UserName" />
                <Parameter name="Password" />
                <Instructions>
                        <Send>(UID "$UserName" "$Password")</Send>
                </Instructions>
        </Init>
        <Method name="SwitchChannel">
                <Parameter name="Channel" />
                <Instructions>
                        <Send>(CHA $Channel)</Send>
                </Instructions>
        </Method>
        <Method name="ShutterOpen">
                <Instructions>
                        <Send>(SHU 0)</Send>
                </Instructions>
        </Method>
        <Method name="ShutterClose">
                <Instructions>
                        <Send>(SHU 1)</Send>
                </Instructions>
        </Method>
</StreamDevice>
```

5.4. Customizing “AutomationCues.xml” for a DLP Cinema Projector

Once the projector file has been configured in “AutomationDevices.xml”, the Cues associated with this can be added to “AutomationCues.xml”. The following is a sample for the Christie projector that provides Automation Cues to change Channels as well as to Open and Close the Projector’s Shutter:

```
<?xml version="1.0" encoding="utf-8"?>
<Cues xmlns="http://schemas.qubecinema.com/Automation/Cues/2008-01-26">
        <Cue name="MXFI YCbCr Flat">
                <Actions>
                        <InvokeMethod name="ShutterClose" device="CP2000ZX"/>
                        <InvokeMethod name="Pause" device="Me"/>
                        <InvokeMethod name="SwitchChannel" device=" CP2000ZX">
                                <Parameter name="Channel" value="101"/>
                        </InvokeMethod>
                        <Sleep duration="3" />
                        <InvokeMethod name="ShutterOpen" device="CP2000ZX"/>
                        <InvokeMethod name="Play" device="Me"/>
                </Actions>
        </Cue>
        <Cue name="MXFI YCbCr Scope">
                <Actions>
                        <InvokeMethod name="ShutterClose" device="CP2000ZX"/>
                        <InvokeMethod name="Pause" device="Me"/>
                        <InvokeMethod name="SwitchChannel" device="CP2000ZX">
                                <Parameter name="Channel" value="102"/>
                        </InvokeMethod>
                        <Sleep duration="3" />
                        <InvokeMethod name="ShutterOpen" device="CP2000ZX"/>
                        <InvokeMethod name="Play" device="Me"/>
                </Actions>
        </Cue>
        <Cue name="DCDM XYZ Scope">
                <Actions>
                        <InvokeMethod name="ShutterClose" device="CP2000ZX"/>
                        <InvokeMethod name="Pause" device="Me"/>
                        <InvokeMethod name="SwitchChannel" device="CP2000ZX">
                                <Parameter name="Channel" value="104"/>
                        </InvokeMethod>
                        <Sleep duration="3" />
                        <InvokeMethod name="ShutterOpen" device="CP2000ZX"/>
                        <InvokeMethod name="Play" device="Me"/>
                </Actions>
        </Cue>
        <Cue name="DCDM XYZ Flat">
                <Actions>
                        <InvokeMethod name="ShutterClose" device="CP2000ZX"/>
                        <InvokeMethod name="Pause" device="Me"/>
                        <InvokeMethod name="SwitchChannel" device="CP2000ZX">
                                <Parameter name="Channel" value="103"/>
                        </InvokeMethod>
                        <Sleep duration="3" />
                        <InvokeMethod name="ShutterOpen" device="CP2000ZX"/>
                        <InvokeMethod name="Play" device="Me"/>
                </Actions>
        </Cue>
        <Cue name="Shutter Open">
```

```
                <Actions>
                        <InvokeMethod name="ShutterOpen" device="CP2000ZX"/>
                </Actions>
        </Cue>
        <Cue name="Shutter Close">
                <Actions>
                        <InvokeMethod name="ShutterClose" device="CP2000ZX"/>
                </Actions>
        </Cue>
</Cues>
```

6. Configuring Automation for a Lighting Controller

To enable automation of a lighting controller, it first needs to be configured in the "AutomationDevices.xml" file as described earlier. Once this is completed, the Cues for the Lighting Controller have to be defined in "AutomationCues.xml". The following uses the example of a Lutron RS-232C Lighting Controller that is provided as standard and will be present in the Qube Program Folder.

Commands supported by Lighting Controller's Automation Files

There is only one Action supported by the Lutron Lighting Controller and that is:

ChangeScene(SceneNumber): SceneNumber is the custom Parameter that specifies the scene number that the Lutron Lighting Controller must switch to.

6.1. Sample Lutron ASCII Automation Device Definition file

The following is an example of the Lutron Lighting Controller automation definition xml file:

```
<?xml version="1.0" encoding="utf-8" ?>
<StreamDevice name="Lutron"
xmlns="http://schemas.qubecinema.com/Automation/StreamDevice/2008-01-26">
        <Method name="ChangeScene">
                <Parameter name="SceneNumber" />
                <Instructions>
                        <Send>:A$SceneNumber</Send>
                </Instructions>
        </Method>
</StreamDevice>
```

6.2. Example for customization of “AutomationCues.xml” for a Lighting Controller

Once the standard file has been configured, we are ready to customize the cues to be added for the lighting controller. This is to be done on the file AutomationCues.xml, which is found in the Qube Program Files Folder. The configuration of AutomationCues.xml is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<Cues xmlns="http://schemas.qubecinema.com/Automation/Cues/2008-01-26">
        <Cue name="lights low">
                <Actions>
                        <InvokeMethod name="ChangeScene" device="Lutron">
                                <Parameter name="SceneNumber" value="3" />
                        </InvokeMethod>
                </Actions>
        </Cue>
        <Cue name="lights mid">
                <Actions>
                        <InvokeMethod name="ChangeScene" device="Lutron">
                                <Parameter name="SceneNumber" value="2" />
                        </InvokeMethod>
                </Actions>
        </Cue>
        <Cue name="lights high">
                <Actions>
                        <InvokeMethod name="ChangeScene" device="Lutron">
                                <Parameter name="SceneNumber" value="1" />
                        </InvokeMethod>
                </Actions>
        </Cue>
</Cues>
```

7. General Purpose Input/Output

The GPIO (General Purpose Input/Output) interface can currently be used to control the state of multiple General Purpose Output lines from the DC20 server. Using GPIO, traditional film automation systems can be controlled in flexible manner, thus facilitating the use of a DC20 system in a film projection environment.

7.1. Setup of the GPIO USB Device

The “Elexol” GPIO Automation Device has one input and one output port to control devices. These ports are represented by Port Numbers 0 and 1. 0 is an input, 1 is an output.

A port is configured as an Input Port by adding that port to *InputPorts* as shown in the example below.

```
<Key name="InputPorts" value="0" />
```

A port is configured as an Output Port by adding that port to *OutputPorts* as shown in the example below.

```
<Key name="OutputPorts" value="1" />
<Key name="DefaultOutputState" value="OFF" />
```

For Output Ports, a default state can be configured using *DefaultOutputState*. The allowed values are ON, OFF or CURRENT:

ON – the corresponding output port will be ON when this Elexol device is loaded

OFF – the corresponding output port will be OFF when this Elexol device is loaded

CURRENT – the corresponding output port will not be changed

In the example below, Port 1 is configured as Output Port, and the default output state is OFF.

```
<Device name="Elexol " class="Qube.Automation.Elexol.Serial, Elexol" enable="true">
        <Configuration>
                <Key name="File" value="Elexol.dll" />
                <Key name="Settings" value="COM4,9600,n,8,1" />
                <Key name="InputPorts" value="0" />
                <Key name="OutputPorts" value="1" />
                <Key name="DefaultOutputState" value="OFF " />
        </Configuration>
</Device>
```

7.2. “Elexol” Device Actions

- **SetPin(Port, Pin, State)**

This Action sets the specified Pin in the specified Port to the specified State

Port: Output Port Number (0 or 1). This Custom Parameter is optional and will default to 1

Pin: Specific Pin Number within a Port where the State is to be changed (0 to 7). It can be a single Pin or a comma separated set of multiple Pins

State: The State to which the Pin must be set (Either ON or OFF)

- **SetPort(Port, State)**

This Action sets all the Pins in the specified Port to the specified State

Port: Output Port Number (0 or 1). This Custom Parameter is optional and will default to 1

State: The State to which the Pin must be set (Either ON or OFF)

- **TogglePin(Port, Pin)**

This Action toggles the existing State of the specified Pins

Port: Output Port Number (0 or 1). This Custom Parameter is optional and will default to 1

Pin: Specific Pin Number within a Port where the State is to be changed (0 to 7). It can be a single Pin or a comma separated set of multiple Pins

7.3. Customization of “AutomationCues.xml” for GPIO using the “Elexol” Device

The Cues to be added for the “Elexol” Device can be customized in “AutomationCues.xml”. The following is an example:

```
<?xml version="1.0" encoding="utf-8" ?>
<Cues xmlns="http://schemas.qubecinema.com/Automation/Cues/2008-01-26">
        <Cue name="gpo-0-on">
                <Actions>
                        <InvokeMethod name="SetPin" device="Elexol">
                                <Parameter name="Pin" value="0" />
                                <Parameter name="State" value="ON" />
                        </InvokeMethod>
                </Actions>
        </Cue>
        <Cue name="gpo-123-on">
                <Actions>
                        <InvokeMethod name="SetPin" device="Elexol">
                                <Parameter name="Pin" value="1,2,3" />
                                <Parameter name="State" value="ON" />
                        </InvokeMethod>
                </Actions>
        </Cue>
        <Cue name="gpo-all-On">
                <Actions>
                        <InvokeMethod name="SetPort" device="Elexol">
                                <Parameter name="State" value="ON" />
                        </InvokeMethod>
                </Actions>
        </Cue>
        <Cue name="gpo-all-Off">
                <Actions>
                        <InvokeMethod name="SetPort" device="Elexol">
                <Parameter name="State" value="OFF" />
            </InvokeMethod>
        </Actions>
    </Cue>
    <Cue name="gpo_toggle-01">
        <Actions>
            <InvokeMethod name="TogglePin" device="Elexol">
                <Parameter name="Pin" value="0,1" />
            </InvokeMethod>
        </Actions>
    </Cue>
    <Cue name="gpo_all_On_Port1">
        <Actions>
            <InvokeMethod name="SetPort" device="Elexol">
                <Parameter name="State" value="ON" />
            </InvokeMethod>
        </Actions>
    </Cue>
    <Cue name="gpo_all_Off_Port1">
        <Actions>
            <InvokeMethod name="SetPort" device="Elexol">
                <Parameter name="State" value="OFF" />
            </InvokeMethod>
        </Actions>
    </Cue>
</Cues>
```

8. Qube ASCII Control Protocol

The DC20 server can be controlled by a simple, yet powerful ASCII command set over IP and through a Serial port. Using this control protocol, the DC20 server can be quickly interfaced with external automation systems as a slave.

And by setting up the automation of one DC20 system to control another through the ASCII protocol, a single DC20 system can also control other DC20 systems in a master/slave configuration.

8.1. Configuring the ASCII Protocol over IP and Serial Port

By default, the DC20 system is setup to respond to ASCII Protocol commands over IP on Port 5000 and over the Serial port COM1 at 19200 baud with no parity, 8 data bits and 1 stop bit (COM1,19200,n,8,1). This can be modified by editing the Dalapathi.exe.config file in the Qube Program Folder. The following Keys in this xml file need to be modified as necessary.

```
<configuration>
        <appSettings>
                <add key=”ASCIIProtocolTCP” value=”5000”/>
                <add key=”ASCIIProtocolSerial” value=”COM1,19200,n,8,1”/>

        </appSettings>
</configuration>
```

8.2. ASCII Protocol Commands

Emergency Stop (ESTP)	
DESCRIPTION Perform an emergency stop on the server.	
SUBCODE None	**DESCRIPTION OF USE** None
EXAMPLES (ESTP)	

Load Show (LOAD)	
DESCRIPTION Load the specified show.	
SUBCODE None	**DESCRIPTION OF USE** Load Show (LOAD “show name”)
EXAMPLES (LOAD “Dark Knight”) Load the show named “Dark Knight”	

Pause Show (PAUS)	
DESCRIPTION Pause the currently playing show.	
SUBCODE None	**DESCRIPTION OF USE** None
EXAMPLES (PAUS)	

<table>
<tr><th colspan="2">Play Show (PLAY)</th></tr>
<tr><td colspan="2"><b>DESCRIPTION</b>
Play the currently loaded show.</td></tr>
<tr><td><b>SUBCODE</b>
None</td><td><b>DESCRIPTION OF USE</b>
None</td></tr>
<tr><td colspan="2"><b>EXAMPLES</b>
(PLAY)</td></tr>
</table>

<table>
<tr><th colspan="2">Resume Show (RSME)</th></tr>
<tr><td colspan="2"><b>DESCRIPTION</b>
Resume playing the currently loaded show from the point it was interrupted when it was last played.</td></tr>
<tr><td><b>SUBCODE</b>
None</td><td><b>DESCRIPTION OF USE</b>
None</td></tr>
<tr><td colspan="2"><b>EXAMPLES</b>
(RSME)</td></tr>
</table>

<table>
<tr><th colspan="2">Login (USER)</th></tr>
<tr><td colspan="2"><b>DESCRIPTION</b>
Log in the specified user. Except for the (VRSN) function, all other functions require a valid user to be logged in with this function.</td></tr>
<tr><td><b>SUBCODE</b>
None</td><td><b>DESCRIPTION OF USE</b>
Log in user (USER “username” “password”)</td></tr>
<tr><td colspan="2"><b>EXAMPLES</b>
(USER “Projectionist” “LetMeIn”) Log in the user “Projectionist” with the password “LetMeIn”</td></tr>
</table>

<table>
<tr><th colspan="2">Get Version (VRSN)</th></tr>
<tr><td colspan="2"><b>DESCRIPTION</b>
Get the version of the ASCII Protocol.</td></tr>
<tr><td><b>SUBCODE</b>
None</td><td><b>DESCRIPTION OF USE</b>
None</td></tr>
<tr><td colspan="2"><b>EXAMPLES</b>
(VRSN?)(VRSN!2.1.3.1)</td></tr>
</table>

<table>
<tr><th colspan="2">LogOut (USER)</th></tr>
<tr><td colspan="2"><b>DESCRIPTION</b>
Log out the user that has been logged in.</td></tr>
<tr><td><b>SUBCODE</b>
None</td><td><b>DESCRIPTION OF USE</b>
None</td></tr>
<tr><td colspan="2"><b>EXAMPLES</b>
(USER)</td></tr>
</table>