Newsroom

Integration and workflows

DATE	28/02/2017
PRINCIPAL AUTHOR	S KIRWAN
SECONDARY AUTHORS	
VERSION	V1.0
UPDATE	
MISC	Docs style applied Dec 2020

Overview

This document describes the tOG Newsroom product. tOG Newsroom brings together several RTSW products into a cohesive whole aimed at the fast creation, completion and transmission of graphic templates in the context of a standard newsroom.

This document sets out the overall configuration and the constituent applications of the product. It follows the life-cycle of a graphic template from creation to transmission - from the creation of graphic templates by the design department through the completion of those templates by journalists at their desktops finishing in the transmission of the completed templates manually in a gallery or using automation systems and the MOS protocol.

Related Manuals

- tOG 2D Edit
- tOG 3D Edit
- tOG Repository
- tOG NewsRoom (this manual)
- tOG Render
- tOG MediaWatcher
- tOG DataServer
- tOG WebControlDesk
- tOG Playout
- tOG Live

The tOG NewsRoom manual covers the applications tOG ENPSActiveX, tOG MOSGateway, tOG MOSObjectEditor and tOG MCAVX2Plugin.

Integration with SAM

The tOG Newsroom solution integrates directly with two SAM products.

The RT applications (tOG WebControlDesk and tOG Playout) can control the RT DLL embedded in SAM ICE automated playout platform.

The RT DLL is also integrated into SAM Rio. Clips on the Rio timeline can be configured to

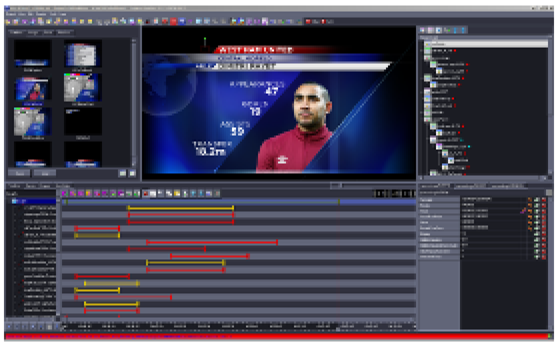
Applications

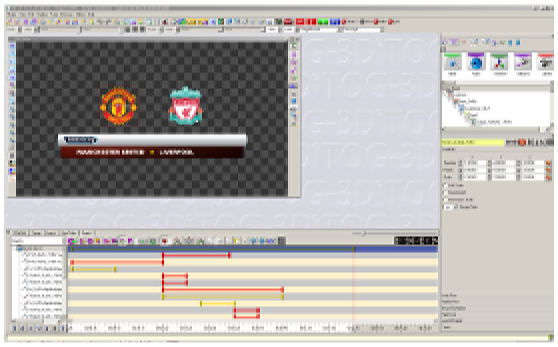
This section lists and describes the applications used in the tOG Newsroom product. These applications can be used in smaller subsets (for example, tOG Edit2D and tOG Playout can be used together as a Character Generator - a simple low cost platform for creating and playing out graphics) but work together in a distributed fashion to provide newsroom functionality.

The descriptions below are cursory and meant as a guide. Please refer to the manuals for the individual applications for a full description of the application’s features and functionality.

Graphic Creation

- tOG Edit2D
- tOG Edit3D



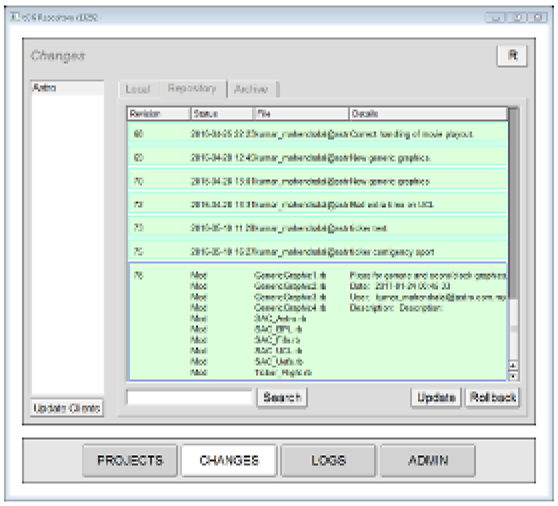


These are the standard RTSW tools for creating and authoring graphics. The final outcome from using these tools is a graphics project. This is a folder which contains all the assets used by the graphics (font, images, geometries etc.) and the scripts and manifests that encapsulate the structure and behaviour of the graphics.

tOG Edit2D handles 2D graphics with simpler structure and more limited data requirements. tOG Edit3D is the full editor and can be used to create any form of broadcast graphic of any complexity.

Graphic and Asset Distribution

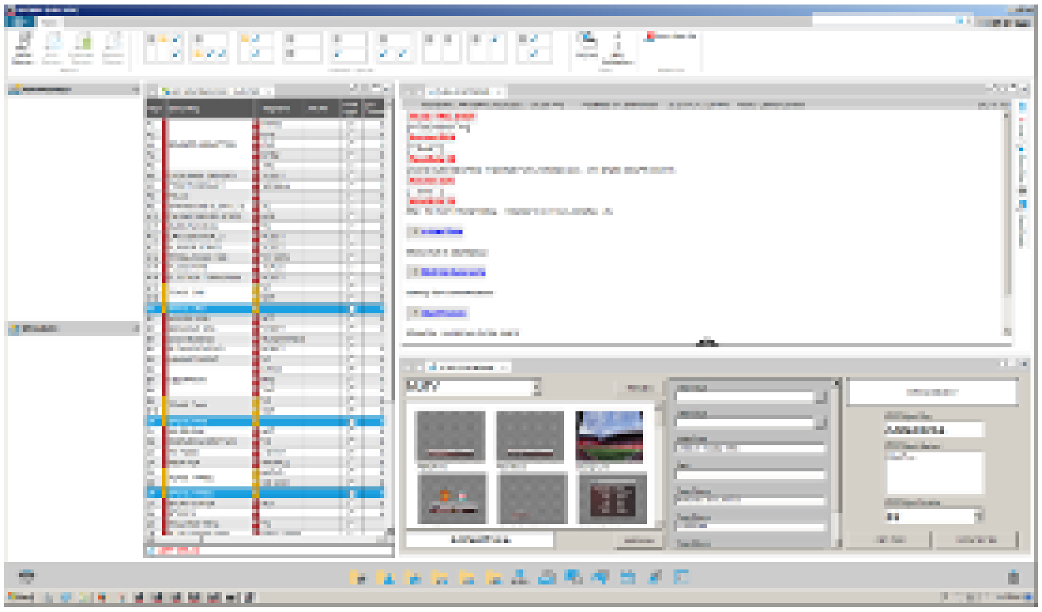
- tOG Repository
- tOG SyncAssets



tOG Repository is used to maintain a “versioned database” of graphic templates (and associated assets) where each change is logged and to distribute the latest version of these graphics to an editable list of client machines - in the newsroom case this would be the central data server and all the rendering machines.

Template Completion & NewsRoom Desktop Clients

- tOG iNewsActiveX
- tOG ENPSActiveX
- tOG OpenMediaActiveX



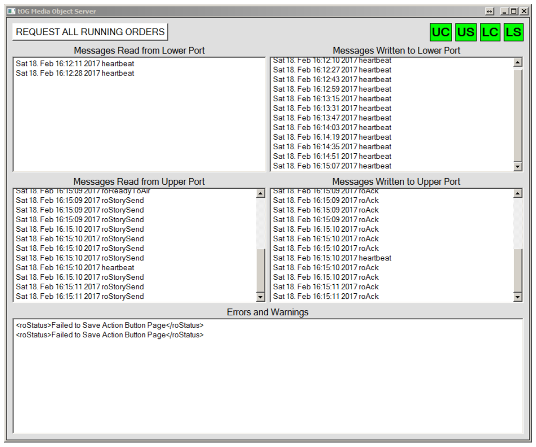
tOG NCS ActiveX implements the ActiveX functionality for NCS desktop clients as described in the MOS Protocol. The purpose of these controls is to enable a non-expert user (for example, a journalist) to add graphic templates to a story in an NCS running order.

These ActiveX controls allow users to select a graphic template, complete the template (by entering values for template parameters), preview the completed graphic template and then add it to a story as a new item. The user can specify the objSlug and mosAbstract for the item.

RT maintains separate ActiveX controls for each of the supported NCS desktop clients.

NCS Interface

tOG MOSGateway



Messages Read from Lower Port

```
Sat 18. Feb 16:12:11 2017 heartbeat
Sat 18. Feb 16:12:28 2017 heartbeat
```

Messages Written to Lower Port

```
Sat 18. Feb 16:12:10 2017 heartbeat
Sat 18. Feb 16:12:27 2017 heartbeat
Sat 18. Feb 16:12:43 2017 heartbeat
Sat 18. Feb 16:12:59 2017 heartbeat
Sat 18. Feb 16:13:15 2017 heartbeat
Sat 18. Feb 16:13:31 2017 heartbeat
Sat 18. Feb 16:13:47 2017 heartbeat
Sat 18. Feb 16:14:03 2017 heartbeat
Sat 18. Feb 16:14:19 2017 heartbeat
Sat 18. Feb 16:14:35 2017 heartbeat
Sat 18. Feb 16:14:51 2017 heartbeat
Sat 18. Feb 16:15:07 2017 heartbeat
```

Messages Read from Upper Port

```
Sat 18. Feb 16:15:09 2017 roReadyToAir
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
```

Messages Written to Upper Port

```
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
```

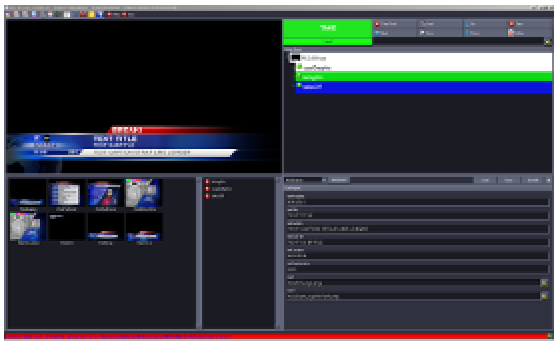
Errors and Warnings

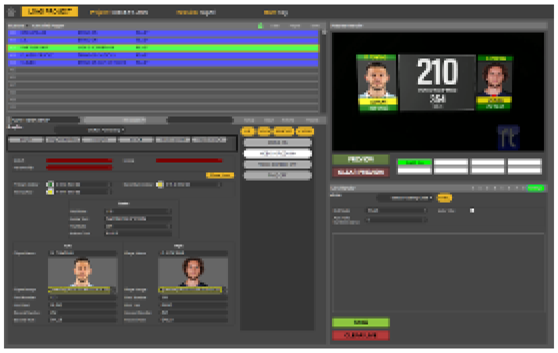
```
<roStatus>Failed to Save Action Button Page</roStatus>
<roStatus>Failed to Save Action Button Page</roStatus>
```

tOG MOSGateway communicates with the Newsroom Control System. It uses the standard MOS protocol to query the NCS about the status and contents of the running orders compiled using the NCS. It also handles MOS messages from the NCS reflecting changes to the current running order - especially the addition of graphics. It creates graphic stacks and pages for later playout from the running orders it receives from the NCS.

Gallery Transmission

- tOG Playout
- tOG WebControlDesk (using tOG DataServer and tOG Render)
- tOG Live





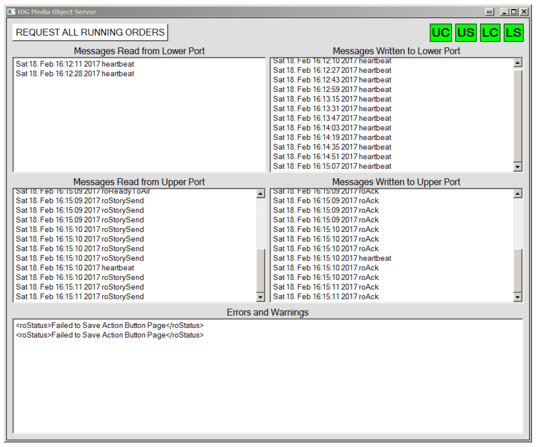
tOG WebControlDesk is a HTML5 application server by tOG DataServer. It can load the playlists and pages create by tOG MOSGateway. tOG Render is a lightweight version of tOG Live which outputs a transport stream. This is used for preview in the web control application.

The tOG WebControlDesk application is accessible from any desktop (that has network access to the server). Playlists or action button pages of graphic templates can be compiled and previewed at the desktop - these playlists and pages are automatically available to operators for transmission. This route to transmission by-passes the NCS completely.

tOG Live is the full RT renderer and outputs SDI video. It is controlled by either the tOG WebControlDesk (using the Web Control Protocol) or tOG Playout using the Render Protocol.

Automated Transmission

- tOG MOSGateway
- tOG Live



Messages Read from Lower Port

```
Sat 18. Feb 16:12:11 2017 heartbeat
Sat 18. Feb 16:12:28 2017 heartbeat
```

Messages Written to Lower Port

```
Sat 18. Feb 16:12:10 2017 heartbeat
Sat 18. Feb 16:12:27 2017 heartbeat
Sat 18. Feb 16:12:43 2017 heartbeat
Sat 18. Feb 16:12:59 2017 heartbeat
Sat 18. Feb 16:13:15 2017 heartbeat
Sat 18. Feb 16:13:31 2017 heartbeat
Sat 18. Feb 16:13:47 2017 heartbeat
Sat 18. Feb 16:14:03 2017 heartbeat
Sat 18. Feb 16:14:19 2017 heartbeat
Sat 18. Feb 16:14:35 2017 heartbeat
Sat 18. Feb 16:14:51 2017 heartbeat
Sat 18. Feb 16:15:07 2017 heartbeat
```

Messages Read from Upper Port

```
Sat 18. Feb 16:15:09 2017 roReadyToAir
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
```

Messages Written to Upper Port

```
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
```

Errors and Warnings

```
<roStatus>Failed to Save Action Button Page</roStatus>
<roStatus>Failed to Save Action Button Page</roStatus>
```

Automated newsroom systems can control tOG Live renders indirectly. They do this by sending the MOS Protocol roCtrl messages to the tOG MOSGateway.

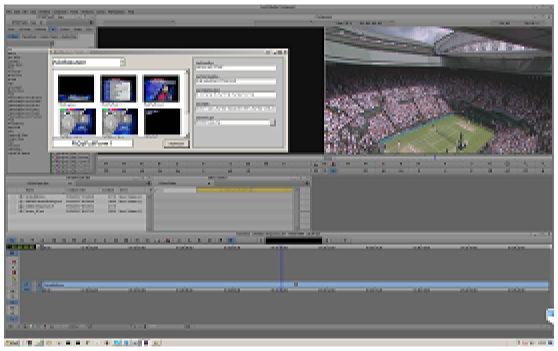
This message commands tOG MOSGateway to playout an item from a story in a running order (the running order, story and item are specified in the message). The item will have been created by dragging from a tOG NCSActiveX embedded in a NCS Desktop Client onto a story in a running order. The graphic template and parameters

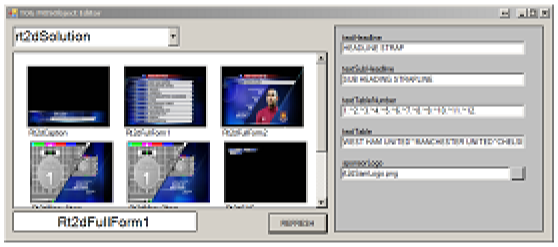
The graphics that have implemented the cueGraphic, bringOn and takeOff methods. These map to the READY, EXECUTE and STOP commands in the MOS message.

The url of the tOG Live to control is specified in the tOG MOSGateway .conf file. Only one tOG Live instance can be controlled by a tOG MOSGateway instance.

NLE Interface

- tOG MOSObjectEditor
- tOG AvidMCAVX2Plugin
- RT DLL





rt2dSolution

textHeadline: HEADLINE STRAP

textSubHeadline: SUB HEADING STRAPLINE

textTableNumber: 1^2^3^4^5^6^7^8^9^10^11^12

textTable: WEST HAM UNITED^MANCHESTER UNITED^CHELS

sponsorLogo: [illegible]

Rt2dFullForm1

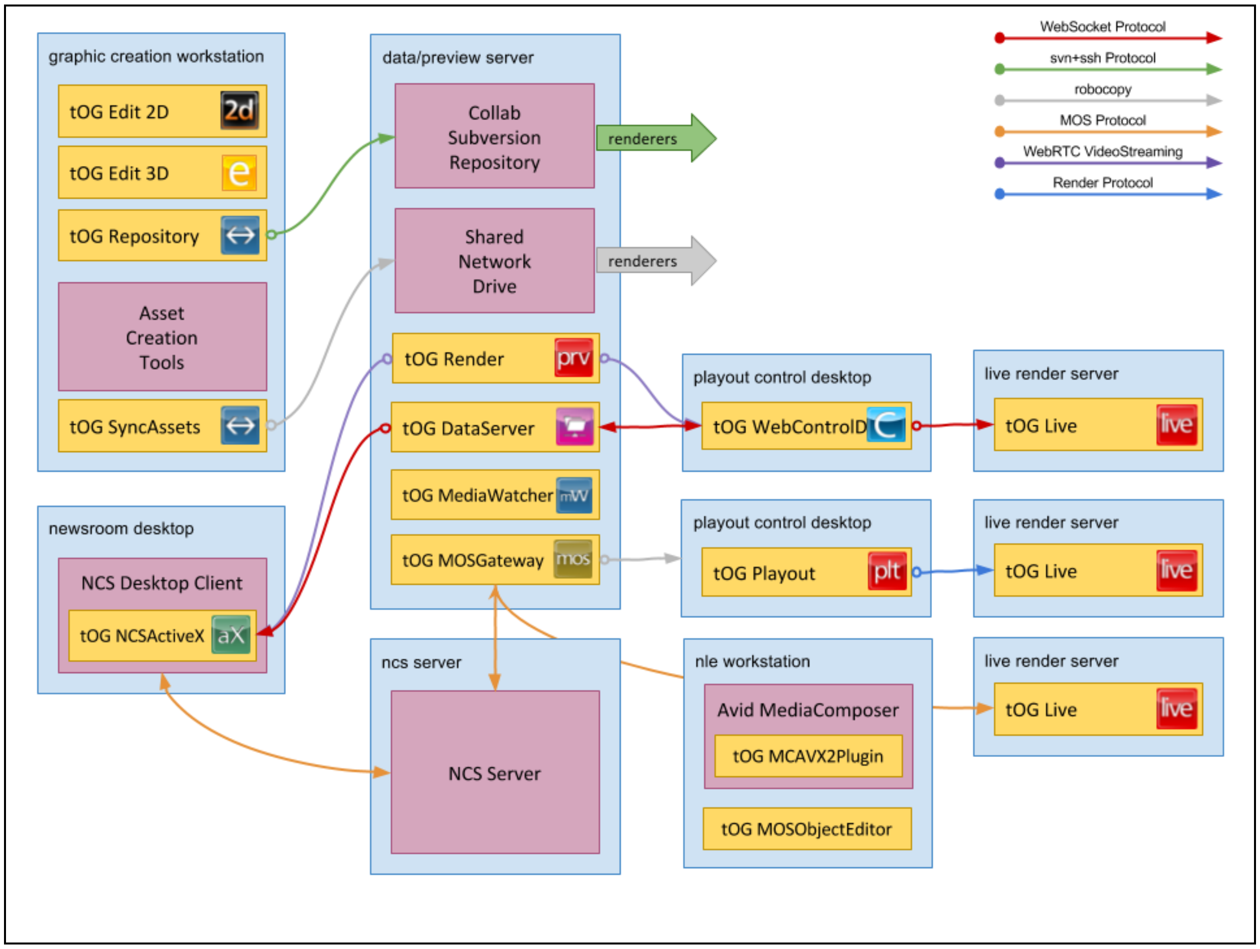
The RT DLL encapsulates the RT render-engine in a dll that can be incorporated into other applications under license. RT has created an AVX2 plugin for Avid MediaComposer that allows the editor to render graphics onto the timeline within the application. It does not rely on an external process or render farm to create the graphics.

The renderer is built into an AVX2 effect which can be dragged onto a timeline clip in the usual way. This causes (in effect mode) the tOG MOSObjectEditor to be run up with the details (if any) of the effect. Within the tOG MOSObjectEditor, the user can select a graphic template and supply values for its parameters. The graphic template must have bringOn and take Off methods. The bringOn method is rendered at the start of the clip and the takeOff at the end.

This graphic will then be rendered onto the clip in real-time. The clip can be dragged, trimmed and scrubbed like any other.

Configuration

The newsroom system is a distributed system. It consists of many disparate applications running on a network of computers. The diagram below sets out the platforms, applications and protocols involved. It does not represent an actual newsroom. There is for instance only one journalist desktop platform - in an actual newsroom there would be many.



Data and control flow in the diagram above moves from left to right. On the left designers, producers and journalists create graphic templates and running orders (which reference the graphic templates). These assets are stored in the servers in the centre which distribute them to the section of playout and render machines on the right.

Protocols

The blue boxes represent computers, yellow boxes are RT applications (most having the RT associated product logo) and the pink boxes are 3rd party applications. The arrows represent data and control flow (the colour varying with the protocol type). The protocols used are:

WebSocket Protocol

This is a communication protocol based on websockets. It was developed by RT Software to support the development of HTML5 control applications. The messages are json packets. As well as the basic communications protocol (heartbeats etc), the WebSocket Protocol includes apis for running graphics, accessing databases, accessing graphic assets and manifests etc.

This protocol is used in three ways:

- To allow tOG DataServer to serve and support tOG the WebControlDesk control application,
- To allow tOG WebControlDesk to control tOG Live (to render graphics to air),
- To allow tOG DataServer to serve graphic project manifests and icons to tOG NCSActiveX so the journalist can select and configure graphics.

svn+ssh Protocol

tOG Repository wraps up an industry standard repository product Collab Subversion. It stores graphic assets in a central repository and distributes these assets to machines that require them. It does this by running svn command directly on the involved machines (using ssh to run svn commands on remote machines).

Robocopy

This is a standard Windows file/folder synchronizing command. It efficiently ensures that two folders on the same or different machines are exactly similar - only copying files that have changed, deleting files that have been deleted in the source.

It is used in the network to move transient assets from the machines on which they are created onto the machines that will use them. It would be possible to store these on a central SAN or Shared Network Drive - but it is most efficient and safe to have them locally on the machines that will use them (the renderers for the most part).

Transient assets that are used for a short period and then deleted. For example, promo movies created for an up-coming programme that are irrelevant once the programme has been broadcast. This is in opposition to graphics templates and their assets (that are maintained and distributed using the svn+ssh Protocol) which have a longer life-span - typically above a year.

The other use for this protocol is to distribute playout stacks derived from NCS running orders by tOG MOSGateway to playout control machines running tOG Playout.

MOS Protocol

MOS is a ten year old, evolving protocol for communications between Newsroom Computer Systems (NCS) and Media Object Servers (MOS) such as Video Servers, Audio Servers, Still Stores, Character Generators, Automation Servers, and Prompters. The MOS Protocol development is supported through cooperative collaboration among equipment vendors, software vendors and end users.

An NCS is a centralised system that enables the creation of running orders of news stories. RT supports three so far (iNews, ENPS and OpenMedia).

It has three uses in the newsroom configuration:

- It allows tOG MOSGateway to communicate with the NCS. tOG MOSGateway uses MOS messages to obtain running orders and to respond to MOS messages from the NCS to update current running orders and MOS objects.
- It allows the tOG NCSActiveX (NCS desktop client plugin) to insert graphic templates selected and configured by the journalist into running order stories as items (which are in turn passed onto tOG MOSGateway).

Render Protocol

This is an RT protocol which shares stacks (and the commands to take items in the stack to air) with controlled machines. It allows tOG Playout to control tOG Live and command it to play graphic templates to air. It is a one-to-many protocol - one tOG Playout can control more than one tOG Live (the protocol was originally used in VR studios).

WebRTC Video Streaming

WebRTC is a free, open project that provides browsers and mobile applications with Real-Time Communications (RTC) capabilities via simple APIs. The RT renderer can create a WebRTC (unbuffered) video stream of the graphics it renders.

The tOG Render application (a lightweight wrap of the full RT renderer) uses it to provide a desktop preview of graphics for tOG WebControlDesk.

The WebSocket Protocol has been extended to support this protocol to facilitate the inclusion of a movie window for preview in an HTM5 application.

Platforms

The platforms involved vary from high-end graphics workstations used to create graphic assets to low-end desktop computers used to host the playout HTML5 application in a Chrome browser. The hardware recommended by RT can be found on the RT website at https://rtsw.co.uk/support_specs/#EnablingTechnology.

The types of platform are:

Graphic creation workstation

The recommended platform is a high-end graphics workstation. Designers will use tools like Photoshop and AfterEffects to create images and movies and Maya and Studio Max to create geometries and shaders. These can be imported into tOG Edit 3D (via the fbx exchange format file) and used to author graphics.

Once ready the graphics can be stored in a central repository (on the data/preview server) and then distributed to the playout and NLE renders (and the data/preview server itself for tOG Render) using tOG Repository.

Newsroom desktop

The recommended platform is a very low-end desktop platform. It is mainly used to run the desktop client of the NCS using which journalists can create running orders, enter their stories and store them in the central database of the NCS (hosted on the NCS server).

These NCS Desktop Clients all have support for MOS Protocol compliant ActiveX controls. The tOG NCSActiveX can be loaded as part of the NCS Client (selected from a list of controls or by clicking on an item in a story). The journalist can then select a graphic template and enter data for it (which can then previewed using a still supplied by a tOG Render on the data/preview server). The template and data can then be dragged onto a story (creating a mosObject). This creates an item within the story with a mosPayload that holds a description of the graphics template and a mosObject reference.

Data/preview server

The hardware platform used for this server will vary depending on the the number of supported graphics/newsroom desktops and playout desktops/servers. The server has four functions:

- To host Collab Subversion Edge repository server - the graphics assets repository. It stores the initial version of a graphics project and then every change (with a comment) made to it. The graphics are also stored locally for use by tOG Render. The repository will grow in size over times as graphics are added and changed. Projects are typically less than a Gigabyte.
- To host the shared network drive that stores transient assets before they are distributed to relevant machines. Since these are transient assets they will eventually be deleted (as part of housekeeping) but individual assets can be large (for example, promo movies can be many Gigabytes in size).
- To host tOG MOSGateway. This application runs in the background without interaction reading, saving and converting running orders and messages from the NCS. This uses very little disk (running orders are typically small).
- To host tOG DataServer and tOG Render. This application run in the background without interaction acting as a web server. It serves the tOG WebControlDesk application and any of it project assets or data requirements.
- To host tOG MediaWatcher. This application processes files and feeds and stores the result in a local database.

NCS server

This is not a platform that is configured or supported by RT. Consult the NCS supplier for specifications of their recommended platform.

Playout control desktop

There are two applications that do playout control and the application used will decide which platform is appropriate.

tOG WebControlDesk

This requires the simplest platform - any pc that can run the Chrome browser. The operator starts the browser and directs it to the tOG DataServer on the data/preview server machine. The operator can then login and select which session to use. This will load the tOG WebControlDesk application and connect it to the tOG Render (for preview) and tOG Live (for transmission) specified in the session. The operator can then select, configure, preview and playout graphics.

tOG Playout

This is usually a high-end graphics workstation. Stacks derived from the running orders in the NCS by tOG MOSGateway (copied to this machine by the synchronisation of transient assets) can be loaded into tOG Playout. Items on the stack can be previewed and then taken to air. The preview is done within tOG Playout and the application (using the MOS Protocol) commands a connected tOG Live to transmit it.

Live render server

The recommended platform is a high-end graphics workstation or server and will usually include a video card.

The graphics templates and associated assets are held locally - distributed here by tOG Repository. Any transient assets (movies and images mainly) needed by playout are copied locally (to ensure speed and security of access).

tOG Live is launched by clicking a desktop icon. It loads all the graphic assets on startup (templates are loaded when required and then cached), opens up SDI video inputs and outputs and any network sockets needed for control.

It then waits for a control application to connect. Under control it will play graphics and methods and render the output to the SDI video out. It can either produce a fill and key output and be keyed onto the transmission stream downstream or it can take the transmission stream as input and render the graphics onto it.

NLE renderer

The recommended platform is a high-end graphics workstation or server. NLEs commonly use a lot of memory and disk. The machine will need a graphics card to render the graphics.

The graphics templates and associated assets are held locally - distributed here by tOG Repository. Any transient assets (movies and images mainly) needed by playout are copied locally (to ensure speed and security of access).

The editor using Avid MediaComposer uses the AVX2 effect tOG AvidMCAVX2 to render graphics on to the other clips on the timeline. The final output will be a movie file of the input clips and graphics.

This could then be moved to a live render server and played out using tOG Live.

tOG MOSGateway

Purpose

To connect to the NCS server and retrieve current running orders compiled by the journalists and containing graphic templates inserted into the running order stories using the tOG NCSActiveX for the NCS Desktop Client. These running orders are then converted into stacks and pages which can be used by RT playout applications (tOG Playout and tOG WebControlDesk).

Operation

The application is runup from a desktop icon. It reads all its parameters from a configuration file (see below) and connects to the NCS using the url specified. It creates four sockets (server/client for the upper/lower ports) and indicates the status of the connections on the interface (using grey/green/red buttons). The roReqMachInfo is sent to the NCS.

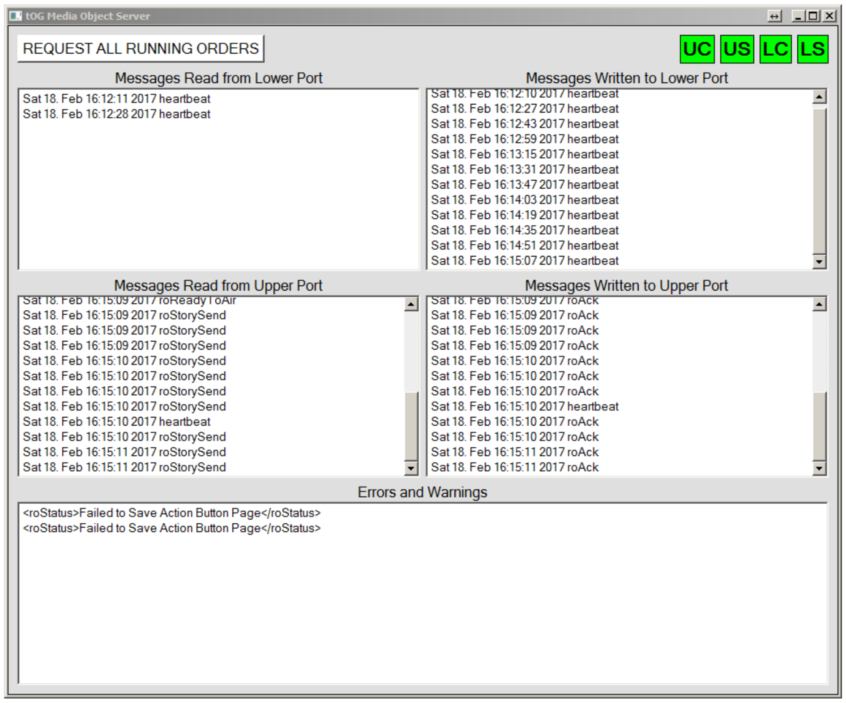
The application heartbeats on all the sockets every 30 secs (if there is no other message). The NCS will send messages when running orders become current, are modified, delete or set as ready-to-air. The application reads, processes and replies to these messages. They are saved to log files in <homeDirectory>/logs/TOGMOSGateway.log and listed on the interface.

The application maintains (on the data/preview server) a local folder of running orders in <homeDirectory>/TOGMOSGateway/runningOrders and another folder of MOS objects in <homeDirectory>/TOGMOSGateway/mosObjects.

These are also converted to stacks and pages and saved locally every time a running order is changed (to <homeDirectory>/projects/<projectName>/GMScripts/Stacks and <homeDirectory>/sessions/<sessionName>/AppData/TOGWebContolDesk/pages). The stacks are then synchronised to playout control desktops running tOG Playout. The pages are served to playout control desktops running tOG WebControlDesk using tOG DataServer.

The application also processes roCtrl messages. The tOG MOSGateway finds the running order, story and item specified in the message (from the files in the runningOrders folder) and extracts the mosPayload from the item. This contains the name of a graphic template and a list of parameter values. It then sends messages to a machine hosting a tOG Live with the url specified by the parameter liveHostName to run methods in the graphic (possibly using the parameters). The values for the command tag in the roCtrl message READY, EXECUTE and STOP map to cueGraphic, bringOn and takeOff methods.

Interface



Messages Read from Lower Port

```
Sat 18. Feb 16:12:11 2017 heartbeat
Sat 18. Feb 16:12:28 2017 heartbeat
```

Messages Written to Lower Port

```
Sat 18. Feb 16:12:10 2017 heartbeat
Sat 18. Feb 16:12:27 2017 heartbeat
Sat 18. Feb 16:12:43 2017 heartbeat
Sat 18. Feb 16:12:59 2017 heartbeat
Sat 18. Feb 16:13:15 2017 heartbeat
Sat 18. Feb 16:13:31 2017 heartbeat
Sat 18. Feb 16:13:47 2017 heartbeat
Sat 18. Feb 16:14:03 2017 heartbeat
Sat 18. Feb 16:14:19 2017 heartbeat
Sat 18. Feb 16:14:35 2017 heartbeat
Sat 18. Feb 16:14:51 2017 heartbeat
Sat 18. Feb 16:15:07 2017 heartbeat
```

Messages Read from Upper Port

```
Sat 18. Feb 16:15:09 2017 roReadyToAir
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:09 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
Sat 18. Feb 16:15:11 2017 roStorySend
```

Messages Written to Upper Port

```
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:09 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 heartbeat
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:10 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
Sat 18. Feb 16:15:11 2017 roAck
```

Errors and Warnings

```
<roStatus>Failed to Save Action Button Page</roStatus>
<roStatus>Failed to Save Action Button Page</roStatus>
```

Request All Running Orders Button

This button will request all currently open running orders from the NCS - it sends a roReqAll command. This is answered with an roListAll message from the NCS. The application then sends a roReq for each running order listed in the roListAll message. The NCS then replies with an roList message containing the full details of the running order which is saved to disk.

UC Button/US Button/LC Button/LS Button

These buttons indicated the status of the four sockets the application opens with the NCS - white for unconnected, green for connected and red for disconnection.

- Messages Read from Lower Port ListBox
- Messages Read from Upper Port ListBox
- Messages Written to Lower Port ListBox
- Messages Written to Upper Port ListBox

All messages sent to the NCS or received from the NCS are listed here with the time and type of command. The full message is written to the log file.

Configuration File

The file is called TOGMOSGateway.conf and located at:

- C:/Users/tog/Desktop/RTSW/conf.

homeDirectory

This is C:/Users/tog/Desktop/RTSW by default. It contains the TOGMOSGateway folder. All the running orders and MOS objects

mosID

The ID of the MOS gateway - rtsw.togmosgateway.fulham.rtsw.mos.

ncsID

The ID of the NCS.

ncsHostName

The ip address or url of the machine hosting the NCS server.

liveHostName

The ip address of the machine hosting a tOG Live to be controlled by the NCS via the MOS Gateway to playout graphics to air.

sessionName

The name of the session (from the tOG WebControlDesk) to which action button pages derived from running orders will be saved.

projectName

The name of the project to which stacks derived from running orders will be saved.

Profile Support

The MOS Gateway supports MOS Protocol v2.8.4.

Profile 0

- heartbeat - heartbeating on all sockets.
- reqMachInfo - returns listMachInfo.
- listMachInfo - used to reply to a reqMachInfo from the NCS.

Profile 1

- mosReqObj - read the the mosObject file and return if it exists; otherwise mosAck.
- mosObj - used to reply to a mosReqObj from the NCS.
- mosReqAll - read the the mosObject files and return in a mosListAll.
- mosListAll - used to reply to a mosReqAll from the NCS.

Profile 2

- roCreate - saves the running order to a file and returns roAck.
- roReplace - saves the running order to a file and returns roAck.
- roDelete - deletes the running order file and returns roAck.
- roReq - returns the running order if the file exists; otherwise roAck with status of NACK.
- roList - saves the running order in the messages to a file.
- roMetadataReplace - unused; returns roAck.
- roElementStat - unused; returns roAck.
- roElementAction - the running order file is modified and saved; returns roAck.
- roReadyToAir - creates a readyToAir file for the running order; returns roAck.

These messages are included in the protocol for backwards compatibility with previous versions of the protocol:

- roStoryAppend - adds the supplied story to the file of the running order specified; return roAck.
- roStoryInsert - inserts the supplied story into the file of the running order specified; return roAck.
- roStoryReplace - replaces the specified story with the supplied story in the file of the running order specified; return roAck.
- roStoryMove
- roStoryMoveMultiple

- roStorySwap
- roItemInsert
- roItemReplace
- roItemMoveMultiple
- roItemDelete
- roStat
- roItemStat

Profile 3

- mosObjCreate - derive a mosObject from the message and save to a file; returns mosAck and the mosObj to the NCS.
- mosItemReplace - unused.
- mosReqObjList - unsupported; returns mosAck.
- mosReqSearchableSchema - unsupported; returns mosAck.
- mosListSearchableSchema - unsupported; returns mosAck.
- mosReqObjList - unsupported; returns mosAck.
- mosObjList - unsupported; returns mosAck.
- mosReqObjAction - unsupported; returns mosAck.

Profile 4

- roReqAll - returns roListAll containing all the running order files.
- roListAll - used to reply to a roReqAll from the NCS.
- roStorySend - replace the story in the running order file; returns roAck.

Profile 5

- roItemCue - unsupported; returns roAck.
- roCtrl - passes on commands to tOG Live on liveHostName; return roAck.

Profile 6

- unsupported.

Profile 7

- roReqStoryAction - unused.

tOG *NCS* ActiveX

Purpose

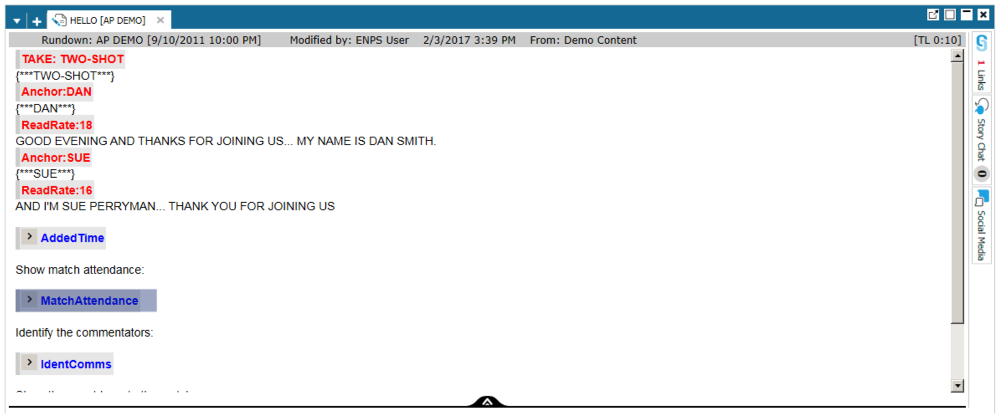
To run inside the NCS Desktop Client used by a journalist. It allows for the selection and configuration of a graphic template and its insertion into a story in the currently open running order.

Operation

From within the NCS Desktop Client the journalist can load the tOG NCS ActiveX - how this is done varies between vendors. The interface appears in a window within the client.

The journalist selects the project to use (usually there is a project for each programme). This will populate the grid of graphic template items. The journalist then selects a graphic template by clicking on it. This will populate the graphic template label and the parameters editor. The journalist can now enter the graphic template details (selecting images etc). The journalist can at any time preview the graphic to check text field widths etc.

All this is creating an ncsItem and once the MOS Object Slug, Abstract and Object Duration the ncsItem is ready to be dragged onto the story.



Rundown: AP DEMO [9/10/2011 10:00 PM]   Modified by: ENPS User   2/3/2017 3:39 PM   From: Demo Content   [TL 0:10]

TAKE: TWO-SHOT
{***TWO-SHOT***}
Anchor:DAN
{***DAN***}
ReadRate:18
GOOD EVENING AND THANKS FOR JOINING US... MY NAME IS DAN SMITH.
Anchor:SUE
{***SUE***}
ReadRate:16
AND I'M SUE PERRYMAN... THANK YOU FOR JOINING US

AddedTime

Show match attendance:

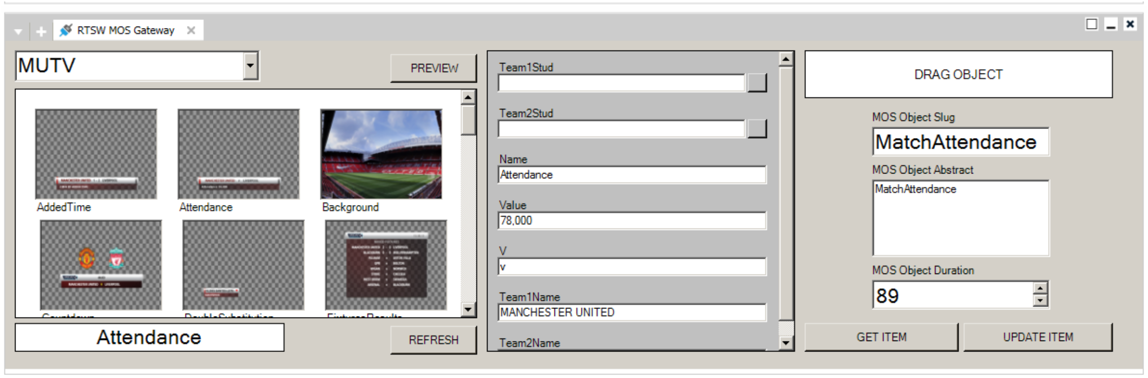
MatchAttendance

Identify the commentators:

IdentComms

The story has to be saved for the item to be sent to the NCS. This will cause a story update (roStorySend) to be sent to the tOG MOSGateway. If the ncsItem is updated by changing the selected graphic template or parameter values, the journalist updates the NCS by clicking the UPDATE ITEM button.

Interface



MUTV

Attendance

Team1Stud

Team2Stud

Name: Attendance

Value: 78,000

V: v

Team1Name: MANCHESTER UNITED

Team2Name

MOS Object Slug: MatchAttendance

MOS Object Abstract: MatchAttendance

MOS Object Duration: 89

As it appears in the ENPS Desktop Client.

<project menu>
A drop-down menu listing all the projects in the <projectsFolder>.

<graphic template icon grid>
Displays the icons for all the graphic templates in the selected project. Click on an icon to select a graphic and populate a parameters editor.

Preview Button
This button will popup a ¼ HD sized images showing a final-frame preview of the selected graphics template with the supplied parameter values.

Refresh Button
This button will re-populate the graphic template icon grid.

<parameters editor>
For each parameter in the graphic (across the cueGraphic, bringOn and takeOff methods), a control is added to the interface (a text box with a image file browser button for TextureImage parameters; otherwise a text box). The journalist can select images or enter data for each parameter.

DragObject Label
Once complete, dragging this label to an open story will insert an item into it by sending an ncsItem to the NCS) - with the MOS Object Slug, Abstract and Duration specified in the

interface and the selected project, graphic template and parameters stored in the mosPayload of the item.

MOS Object Slug
Populates the itemSlug field of the ncsItem dragged to the NCS Desktop Client.

MOS Object Abstract
Populates the mosAbstract and abstract fields of the ncsItem dragged to the NCS Desktop Client.

MOS Object Duration
Populates the itemEdDuration and itemUserTimingDuration fields of the ncsItem dragged to the NCS Desktop Client.

Get Item Button
This button gets the current item from the NCS Desktop Client and populate the interface. It is also possible just to click on the item in the story.

Update Item Button
This button updates the open item with new parameters values entered into the interface by the journalist.

Configuration File
The file is called TOGMOSGateway.conf and located at C:/RTSW.

ncsID
The ID of the NCS.

mosID
The ID of the MOS.

imagesFolder
The location images to be used in the graphic templates. This will be the Shared Network drive of transient assets.

projectsFolder

The location of the projects listed in the interface. This will be the projects folder checked out from the graphic template repository.

previewImage

The location of the preview image.

previewURL

previewLowerPort

previewUpperPort

The url and ports of the tOG Render application used render the preview image.

- dataServerURL
- dataServerUserName
- dataServerPassword
- dataServerSessionName

The details of tOG DataServer. The application can obtain a list of all projects in the graphic template repository, project manifests, graphic manifests and icons from the tOG DataServer.

tOG MOSObjectEditor

Purpose

To work in concert with the tOG AvidMCAVX2Plugin to configure graphic templates that can be rendered as a clip onto the Avid MediaComposer timeline.

Operation

The application is either launched by tOG AvidMCAVX2Plugin running inside Avid MediaComposer as an effect on a clip if it is not already launched (when the effect is dragged onto a clip on the track or when a clip is clicked on in effect mode) or manually launched by clicking a desktop item.

The editor selects the project to use (usually there is a project for each programme). This will populate the grid of graphic template items. The editor then selects a graphic template by clicking on it. This will populate the graphic template label and the parameters editor. The editor can now enter the graphic template details (selecting images etc). The editor can at any time preview the graphic to check text field widths etc.

Any change (selecting a new graphic or entering new parameter data) will write a file called tOG MOSObjectEditor.mosObj to the folder C:/tOGLogs. This contains all the details of the graphics template configured by the editor as a mosObject.

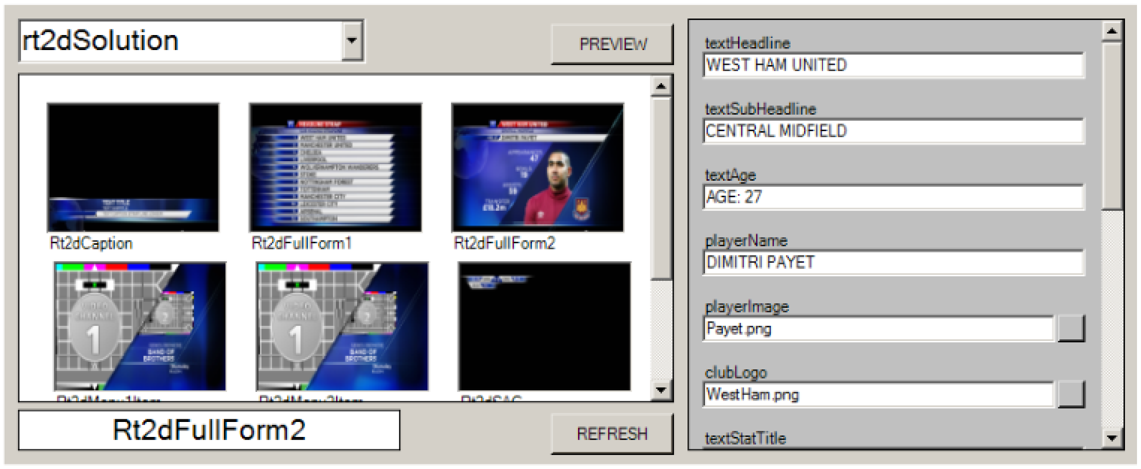
The tOG AvidMCAVX2Plugin will read the file (and then delete it) and renders it to the current clip on the timeline.

Conversely, if a new clip (with a tOG AvidMCAVX2Plugin effect assigned to it) is selected it writes a file called NLE.mosObject to C:/tOGLogs. This contains a mosObject created by the tOG MOSObjectEditor and saved in the Avid MediaComposer project.

tOG MOSObjectEditor reads the file (and deletes it) and then populates the interface with the details.

Interface

The application wraps up the tOG NCS ActiveX.



textHeadline: WEST HAM UNITED

textSubHeadline: CENTRAL MIDFIELD

textAge: AGE: 27

playerName: DIMITRI PAYET

playerImage: Payet.png

clubLogo: WestHam.png

textStatTitle

<project menu>

A drop-down menu listing all the projects in the <projectsFolder>.

<graphic template icon grid>

Displays the icons for all the graphic templates in the selected project. Click on an icon to select a graphic and populate a parameters editor.

Refresh Button

This button will re-populate the graphic template icon grid.

<parameters editor>

For each parameter in the graphic (across the cueGraphic, bringOn and takeOff methods), a control is added to the interface (a text box with a image file browser button for TextureImage parameters; otherwise a text box). The journalist can select images or enter data for each parameter.

Configuration File
The file is called TOGMOSObjectEditor.conf and located at C:/RTSW.

imagesFolder
The location images to be used in the graphic templates. This will be the Shared Network drive of transient assets.

projectsFolder
The location of the projects listed in the interface. This will be the projects folder checked out from the graphic template repository.

tOG AvidMCAVX2Plugin

Purpose

To draw graphics (selected and configured by tOG MOSObjectEditor) directly as a clip onto the timeline inside Avid MediaComposer. This is done by rendering the graphics within editor on the edit machine - not using an external rendering application or render farm.

Operation
tOG AvidMCAVX2Plugin is an Avid AVX2 effect plugin - it wraps the RT renderer as a dynamically loaded dll RT.dll. When installed it appears in the list of effected under RT Software.

The editor drags the effect onto the timeline clip. The effect takes up the whole clip, saves a file C:/tOGLogs/tOGMOSObjectEditor.mosObj (containing a MOS object for an empty graphic template) and launches tOG MOSObjectEditor. The editor can then select and configure a graphic template (see the section on tOG MOSObjectEditor). This is written as a complete MOS object to a file C:/tOGLogs/NLE.mosObj.

The editor then clicks on the timeline clip and this causes the effect to read the MOS object from the NLE.mosObj file and render the graphic template described in the mosPayload field in the MOS object onto the timeline.

There are a few limitations. In any one edit project effects can only use one graphics project - swapping between projects would be prohibitively expensive. Also, there are two noticeable pauses. The first is when the render engine is initialised - this takes a few seconds and occurs when the effect is first selected. It only happens once per session. The second is when the project is loaded. This happens as a result of attempting to render a MOS object if the project specified in it is not already loaded. Again this happens once per session but the time taken depends on the project size. This is mitigated by loading the project on a just-in-time basis but can take a few minutes.

The mosPayload is stored in an effect parameter called EFFECT DESCRIPTION so it will be saved with the project.

The graphic template is expected to contain bringOn and takeOff methods. The bringOn method is rendered at the start of the clip and the takeOff at the end. Behind the scenes the changes to the scenegraph caused by running the bringOn and takeOff methods are captured pre frame (a list of node afields and new values) in a GMSceneGraphUpdate object (one per effect). The frames between the two methods are filled in with the end values from the bringOn method.

The clip can be trimmed and moved - both will cause the GMSceneGraphUpdate object to be re-computed. There can be more than one effect in a project - the graphics will be drawn on top of each other in sequence.

Interface

There is no interface. See the section on tOG MOSObjectEditor.

Configuration File

The file is called TOGAvidMCAVX2Plugin.conf and located at C:/RTSW.

imagesFolder

The location images to be used in the graphic templates. This will be the Shared Network drive of transient assets.

projectsFolder

The location of the projects listed in the interface. This will be the projects folder checked out from the graphic template repository.

Installation

This is done manually. It has three parts:

1. Copy TOGAvidMCAVX2Plugin.avx to the folder C:/Program Files/Avid/Avid Media Composer/AVX2_Plug-Ins.
2. Copy the .dlls (including RT.dll) to C:/Program Files/Avid/Avid Media Composer.
3. Copy the configuration file TOGAvidMCAVX2Plugin.conf to C:/Rtsw.