WASP3D

Automation Protocol 5.0

Overview

WASP3D Automation Protocol lets devices to control the WASP3D sting Server (graphics playout). It is based on Unicode text based commands separated by a carriage return \r and newline character \n.

Automation Systems can connect to Wasp automation driver through TCP/IP. Encoding used for the automation is UTF 8 and commands are case insensitive while parameters are case sensitive.

Order of the parameter needs to be maintained. Since double quote is used as container for Name Value pair. In case the content contains double quote, it needs to be sent with escape character "\". Spaces other than mentioned below will automatically get truncated.

Command<Space>" ParameterName=ParameterValue" <space>" ParameterName=ParameterValue"\r\n

<Space>	Separator used between command and parameters
"	Container for Parameter name and parameter value pair
\r\n	Command terminator

Automation Configurations

WASP3D Automation Protocol is fully configurable depending upon the client working environment. Automation can listen for commands from TCP/IP, UDP, Web socket and Named pipes. This is configurable in the XML of the automation addin as per below parameters.

```
<endpoint port="10005" type="tcp" />
<endpoint port="10006" type="web" />
<endpoint port="10007" type="udp" />
<endpoint port="10008" type="namedpipe" name="listener" />
```

WASP3D Acronyms

Template

A Template (or a Graphic Template) is also known as a Scene graph. It contains the 3d objects along with their animations. A template may contain multiple control keys (e.g. timed pause, infinite pause, play complete etc. which provide the functionality of key frame based actions to be performed in the graphics)

WASP3D Sting Server

This is the real-time rendering engine for the graphics playout. The graphics are delivered as SD/HD – SDI with Fill and Key Outputs.

Player

To Control the Playout of a Graphic Template – The Player (or play control) provides controls to Initialize, Play, Pause, and stop the Graphics Animation.

Playlist

A Playlist (also called rundown) is a collected of graphic template instances that needs to be played. In Wasp3d the data and the template are stored separately. The templates are stored in a central server and the data is stored in the DB. Each playlist instance is a combination (or union) of a template and a data. Data element for a particular template can be created using Wasp3D ActiveX or Data Buzz. When a particular instance is loaded, the system marries the data element with the template and loads them on the Sting Server. The playlist can have predefined "Z" ordering, and hence by default all data instances in the playlist will get loaded in the same 'Z' order. Multiple playlist can be active and played simultaneously at same time on the system.

Program

A Program is container for playlists. Each program can have multiple playlist with different "Z" order. By sending controlling commands to Program, multiple playlist can be controlled at same time. Programs are created using the Wasp3D Hive manager. Once a Program is created users can drag drop existing playlists into this Program or create new playlists and assign them to the program. User can also assign a particular "Z" order for layering purpose.

Data Instance or Template Instance

A Data instance is the data information (of the variable elements of a graphic template) that an end user populates. A Data instance consists of two pieces of information – the Variable Text data that an end user fill in and the associated graphics template for this variable data. Since both these are stored separately – each template can have 'n' number of data instance associated with it.

Change in this version of WASP3D Automation Protocol

WASP3D System support the render of the instance to be played in either Program or Preview modes. Command to Load and Play the scene to the required mode is added as an optional parameter. If the command is sent without this parameter, it would be directed to the Program Mode.

Mode= {Preview / Program} is added in the Load and Play commands in this document. Relevant commands as an example are also added to display the same.

WASP3D Control Commands

Player Commands

Commands in this category are used for controlling individual graphic instances.

Scenarios

When the Automation System owns the rundown and instructs the WASP3D Sting Server to load a specific graphic template (or instance) as per the predefined schedule.

Command Number	**PLR001**
Category	PLAYER
Command	load
Command	Load the specified instance
Compulsory parameters	1

Command	**Parameter 1**	**Optional parameter 1 ***	**Optional parameter 2**	**Optional parameter 3**
load	Instance={Name of the Instance or InstanceID}	Playlist={ Name of the Playlist }	Zorder={value}	Mode={Preview / Program}

Description

This command will load the specified instance into the Sting Server. If the instance doesn't exist, then the command will fail. If there are multiple instances with the same name, then first available instance will be get loaded. If a playlist name is specified, the system will search for the instance in the specified playlist and if found loads that instance. If the Z-order is specified, the instance will be assigned with that number.

If Name of instance is specified, then optionally the playlist name should be passed. If no playlist name is passed along with Instance Name, System will look for that instance in the default playlist configured in the system.

Example

load “Instance= Story 1.wsp”
load “Instance= Story 1.wsp” \r\n
load “Instance= Story 1.wsp” “Playlist=News@9”
load “Instance= Story 1.wsp” “Playlist=News@9” \r\n
load “Instance= Story 1.wsp” “Zorder=1”
load “Instance= Story 1.wsp” “Zorder=1” \r\n
load “Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}”
load “Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}” \r\n //InstanceID
load “Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}” “Playlist=News@9”
load “Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}” “Playlist=News@9” \r\n
load "Instance= db330416-5e76-4721-8412-8a628e9f806a" "Zorder=1"
load "Instance= db330416-5e76-4721-8412-8a628e9f806a" "Zorder=1"\r\n
load "Instance=abc.wspx" "mode=Preview”
load "Instance=abc.wspx" "mode=Program"
load "Instance= db330416-5e76-4721-8412-8a628e9f806a" "mode=Preview" "Zorder=1”
load "Instance= db330416-5e76-4721-8412-8a628e9f806a" "mode=Preview" "Zorder=1" ss="0”

Response

Automation will send a response in Following Format

<Instance ID or Slug>, <Status>
e.g. *Story1, PREPARED*

If playlist is mentioned in command, then response would be in Following Format

<Playlist Name, Playlist id>, <Instance ID or Slug>, <Status>

e.g. *News@9, BAEC83DA-46F5-4C96-AF5A-A6039D55E993, Story1, PREPARED*

Command Number	PLR002
Category	PLAYER
Command	play
Purpose	Start playing the specified instance
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
play	Instance={Name/ID of the Instance}	Mode={Preview / Program}

Description

This command will start playing the specified instance. The Instance should be loaded into the system (see PLR001) before this command. If the specified player is not loaded the command will fail. If the Instance is already loaded and in a "Paused" state (see PLR 003), then it will release the Pause Point.

Example

play "Instance= Story 1.wsp"

play "Instance= Story 1.wsp" \r\n

play "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}"

play "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}" \r\n

Play "Instance=abc.wspx" "mode=Preview"

Play "Instance=abc.wspx" "mode=Program"

Response

Automation will send a response in Following Format

<Instance ID or Slug>, <Status>

e.g. *Story 1.wsp, PLAYING, Frame 0*

Command Number	PLR003
Category	PLAYER
Command	pause
Purpose	pauses specified player
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
pause	Instance={Name/ID of the Instance}	

Description

This command will pause the specified instance. The specified instance should be loaded into the system (see PLR001) before this command. If the specified player is not loaded the command will fail. Once this command is executed, it needs to be released using the play (see PLR002) command.

Example

pause "Instance= Story 1.wsp"

pause "Instance= Story 1.wspx"

pause "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}"

pause "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}" \r\n //Instance ID

Response

Automation will send a response in Following Format

<Instance ID or Slug>, <Status>

e.g. *Story 1.wsp, PAUSE*

Command Number	PLR004
Category	PLAYER
Command	stop
Purpose	stops specified player
Compulsory parameters	1
Optional Parameters	1

Command	Parameter 1	Optional parameter 1
stop	Instance={Name/ID of the Instance}	Graphicout={0 or 1}

Description

This command will stop the specified player. The specified instance should be loaded into the system (see PLR001) before this command. Executing this command will result in stopping of the animation in the graphic and taking the graphic "OFF-AIR". This command will not unload the graphic from memory of play out server until the unload command (see PLR005) is called. A stopped instance can be started again by sending the play command (PLR002) without sending load Player(PLR001) command again. Optional graphic out parameter notify the server to stop the graphic without playing Animation.

Example

stop "Instance= Story 1.wsp"

stop "Instance= Story 1.wspx"

Response

Automation will send a response in Following Format

<Instance ID or Slug>,<Status>

e.g. *Story 1.wsp, STOPPED*

Example

Stop "Instance= Story 1.wsp.wsp" "playoutanimation=1"

Stop "Instance= Story 1.wsp.wspx" "playoutanimation=1"

Stop "Instance= Story 1.wsp.wsp" "playoutanimation=0"

Stop "Instance= Story 1.wsp.wspx" "playoutanimation=0"

stop "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}" "playoutanimation=1"

stop "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}" "playoutanimation=1" \r\n //Instance ID

Response

Automation will send a response in Following Format

<Instance ID or Slug>,<Status>

e.g. *Story 1.wsp,Playing, frame 17 / Story 1.wsp,PAUSEINFINITE / Story 1.wsp,PLAYING,frame no. / Story 1.wsp,PLAYCOMPLETE Story 1.wsp,PAUSE Story 1.wsp, STOPPED*

Command Number	PLR005
Category	PLAYER
Command	unload
Purpose	stop and unload the specified player
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
unload	Instance={Name/ID of the Instance}	

Description

Executing this command will stop the current scenes animation in the graphic and take it "OFF-AIR". The Templates are also unloaded from the Sting Server. The specified instance should be loaded into the system (see PLR001) before this command.

Example

unload "Instance= Story 1.wsp"

unload "Instance= Story 1.wspx"

unload "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}"

unload "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}" \r\n //Instance ID

Response

Automation will send a response in Following Format

<Instance ID or Slug>,<Status>

e.g. *Story 1.wsp,SGDELETED*

Command Number	PLR006
Category	PLAYER
Command	update
Purpose	Updates scene graph variable/ global variable or user tag of the given instance
Compulsory parameters	2
Optional Parameters	0

Command	Parameter 1	Parameter 2
update	Instance={Name/ID of the Instance}	JSON String

Description: Updates scene graph variable/ global variable or user tag of the given instance

Json Format

STRING

update "Instance=variables.wsp" "JSON={"SVar":{"String":[{"Name":"String 1","Value":"New Di"}]}}" \r\n
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"SVar":{"String":[{"Name":"String 1","Value":"New Di"}]}}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"SVar":{"String":[{"Name":"String 1","Value":"New Di"}]}}" \r\n

FLOAT

update "Instance=variables.wsp" "JSON={"SVar":{"Float":[{"Name":"String 1","Value":"New1 Di"}]}}"
update "Instance=variables.wspx" "JSON={"SVar":{"Float":[{"Name":"String 1","Value":"New2 Di"}]}}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"SVar":{"Float":[{"Name":"String 1","Value":"New Di"}]}}"

INTEGER

update "Instance=variables.wsp" "JSON={"SVar":{"Int":[{"Name":"String 1","Value":"New7 Di"}]}}"
update "Instance=variables.wspx" "JSON={"SVar":{"Int":[{"Name":"String 1","Value":"New7 Di"}]}}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"SVar":{"Int":[{"Name":"String 1","Value":"New7 Di"}]}}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"SVar":{"Int":[{"Name":"String 1","Value":"New7 Di"}]}}" \r\n

USER TAG

update "Instance=variables.wsp" "JSON={"UTag":[{"Name":"1","Value":"C","index":"-1"}]}"
update "Instance=variables.wspx" "JSON={"UTag":[{"Name":"1","Value":"C","index":"-1"}]}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"UTag":[{"Name":"1","Value":"C","index":"-1"}]}"
update "Instance=e5b8c629-1d70-4297-afdd-18355057e08b" "JSON={"UTag":[{"Name":"1","Value":"C","index":"-1"}]}" \r\n

Response: Not Applicable

Command Number	PLR007
Category	PLAYER
Command	Play Action set
Purpose	Play the specified action set from the scene
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Parameter 2
play	Instance={Name/ID of the Instance}	actionset={name of the action set in the scene}

Description

This command will play the specified action set from the scene. The Data Instance should be loaded into the system (see PLR001) before this command. If the specified scene is not loaded the command will fail. The scene will play/behave according to the commands and controls specified in the Action Set.

Example

play "Instance= Story 1.wsp""actionset=NAME OF ACTION SET"

play "Instance= Story 1.wspx""actionset=NAME OF ACTION SET"

play "Instance= {BAEC83DA-46F5-4C96-AF5A-A6039D55E993}""actionset=NAME OF ACTION SET" \r\n

Response

Automation will send a response in Following Format

<Action set name>,<Status>

e.g. *Action Set Name, PLAYING*

Command Number	PLR008
Category	PLAYER
Command	Heart Beat
Purpose	Ping
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional Parameter
raise	Instance={Name/ID of the Instance}	

Description

This command is to check whether server is connected or not.

Example

heartbeat "time=07/28/2016 03:44:55”
heartbeat "time=05/11/2015 03:44:55"\r\n

Response

Automation will send a response in Following Format
time=07/28/2015 12:44:55

Command Number	PLR009
Category	PLAYER
Command	Play frame
Purpose	Start playing the specified instance from particular frame number
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
play	Instance={Name/ID of the Instance}	

Description

This command will start playing the specified instance from a particular frame number.

Example

play "Instance= Story 1.wsp.wsp" "framenumber=51"

play "Instance= Story 1.wspx.wsp" "framenumber=5"

play "Instance= 7292a2e1-0d5c-456c-a1e0-2f9e7ff94187" "framenumber=5"

play "Instance= 7292a2e1-0d5c-456c-a1e0-2f9e7ff94187" "framenumber=5" \r\n

Response

Automation will send a response in Following Format

<Instance ID or Slug>, <Status>

Story 1.wsp.wsp, PLAYING , Action set name

Command Number	PLR010
Category	PLAYER
Command	Unload all
Command	unload the specified instance
Compulsory parameters	1

Example

Unload

Unload \r\n

Description

This command will unload all the loaded instances from sting server.

Response

Automation will send a response in Following Format

<Instance ID or Slug>,<Status>

e.g.

Story 1.wspx, SGDELETED

Story 2.wspx, SGDELETED

Story 3.wspx, SGDELETED

Command Number	PLR011
Category	Raise
Command	Raise trigger
Command	Raise a specific trigger
Compulsory parameters	2

A Raise Trigger is used to trigger events applied in the template. For e.g. when a commercial is scheduled, the ticker needs to move out or lower as handled by the TV station. In this case, The Ticker template can have a trigger applied to it so that when a Trigger ID for "Commercial in" is raised, the Ticker is lowered and when the Trigger ID for "Commercial Out" is raised, it comes back to the position.

Any number of triggers can be created, and various templates can use those triggers to perform respectively. Multiple Templates can receive one trigger to work simultaneously as well. Depending on the usage scenario various levels are provided accordingly.

Levels description

- **Internal –** Is raised and received in the same template. > **Level Value = 0**
- **Template –** Is received on the player of same template which has raised the trigger. This trigger can have used by coding in Comb Builder. > **Level Value = 1**
- **Playlist –** Is received on all loaded player(s) of a playlist in which the player of raising trigger is present. This trigger can have used by coding in Comb Builder. > **Level Value = 2**
- **Sting Client –** Is received on all loaded player(s) of all opened playlists of a Sting Client in which the player of raising trigger is present. This trigger can have used by coding in Comb Builder. > **Level Value = 3**
- **Broadcast –** Is received on all loaded player(s) of all opened playlist of all Sting Client(s) connected to the same Kernel Controller. This trigger can have used by coding in Comb Builder. > **Level Value = 4**
- **Sting Server (Active Scene)** - Is received on all the scenes being rendered as on the same server. > **Level Value = 5**
- **Sting Server (All)** - Is received on all the scenes which are loaded on the same server. > **Level Value = 6.**

Raise Trigger Format:

- **Format: Raise "Trigger=<TriggerID>" "Level=<LevelNumber>" "Mode=<RenderMode>"**

Raise Trigger Command >>

Example

- *raise "Trigger=1005" "Level=6" [If nothing is passed in mode parameter then by default it will be triggered on program"]*
- *raise "Trigger=1005" "Level=6"" Mode=Program" [it will trigger on program"]*
- *raise "Trigger=1005" "Level=6"" Mode=Preview" [it will trigger on preview"*

Description

This command will raise trigger at different levels. Triggers are used to initiate actions on player level and on server level.

Response:
automationlog, appname= automationaddin, logtime= 12:48:49.785, severity= Informational, Class= InstanceData, Method= RaiseTrigger, Message= RaiseTrigger for Mode Program/Preview with Trigger id 1005 and Level 6, MachineName= <machine name>, MachineIP= <machine IP>,

Note: *All the responses will be written in a log file after configuration.*

Following are the steps to configure the logs.

Changes in Common config [This config file is a part of WASP installation]
key="WRITE_LOG" value="TRUE"

For log path change path in this key
key="LOGPATH" value="C:\BeeSys\Logs" this is default path

Command Number	PLR011.1
Category	Raise
Command	RaiseNamed
Command	Raise a specific name
Compulsory parameters	2

A Raise Named is used to trigger Named events applied in the template. For e.g. when a commercial is scheduled, the ticker needs to move out or lower as handled by the TV station. In this case, The Ticker template can have a named event applied to it so that when a Named for "Commercial in" is raised, the Ticker is lowered and when the Named for "Commercial Out" is raised, it comes back to the position.

Raise Named Event:

- **Format: RaiseNamed "Named=<Event name>" "Level=<Level>" "Mode=<Render mode>"**

Command:

- *RaiseNamed "Named=<Name>" "Level=6" [If nothing is passed in mode parameter then by default it will be triggered on program"]*
- *RaiseNamed "Named=<Name>" "Level=6" "Mode=Program" [it will trigger on program"]*
- *RaiseNamed "Named=<Name>" "Level=6" "Mode=Preview" [it will trigger on preview"]*

Response:
automationlog, appname= automationaddin, logtime= 12:51:34.739, severity= Informational, Class= InstanceData, Method= RaiseNamedEvent, **Message= RaiseNamed for Mode Program/Preview with Named Event <Name> and Level 6**, MachineName= <machine name>, MachineIP= <machine IP>

Note: *All the responses will be written in a log file after configuration.*

Following are the steps to configure the logs.

Changes in Common config [This config file is a part of WASP installation]
key="WRITE_LOG" value="TRUE"

For log path change path in this key
key="LOGPATH" value="C:\BeeSys\Logs" this is default path

Command Number	PLR012
Category	PLAYER
Command	GetAllScene
Purpose	To Get Information of all the loaded scenes
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
GetAllScene	NA	NA

Description

Example

GetAllScene

Response

<Instance name1>

<Instance name2>

Command Number	PLR013
Category	PLAYER
Command	GetEngineStats
Purpose	To Get Information of Loaded and played scenes with memory information.
Compulsory parameters	1
Optional Parameters	0

Command	Parameter 1	Optional parameter 1
GetAllScene	NA	NA

Description

Example

GetEngineStats

Response

Engines Tats: Current Loaded SG 2, Total Loaded SG 6, Memory Size 191.9492(MB)