PRIME JavaScript User Guide

Version 5.3

January 2026

SINCE DAY ONE.

Chyron PRIME JavaScript User Guide • 5.3 • January 2026 • This document is distributed by Chyron in online (electronic) form only, and is not available for purchase in printed form.

Table of Contents

JavaScript Introduction

Description

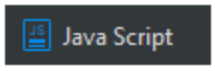
JavaScript is a robust scripting language capable of executing advanced control over multiple scopes; Scene level, Project Level, Application Level, or Master Control Panels.

ECMA Scripting Language

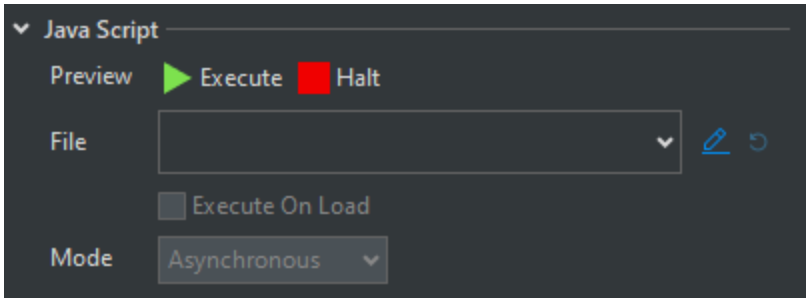
JavaScript component uses the ECMAScript Language standard as defined by the links below.
https://developer.mozilla.org/en-US/docs/Web/JavaScript
https://tc39.es/ecma262/

Java Script Effect and Resource

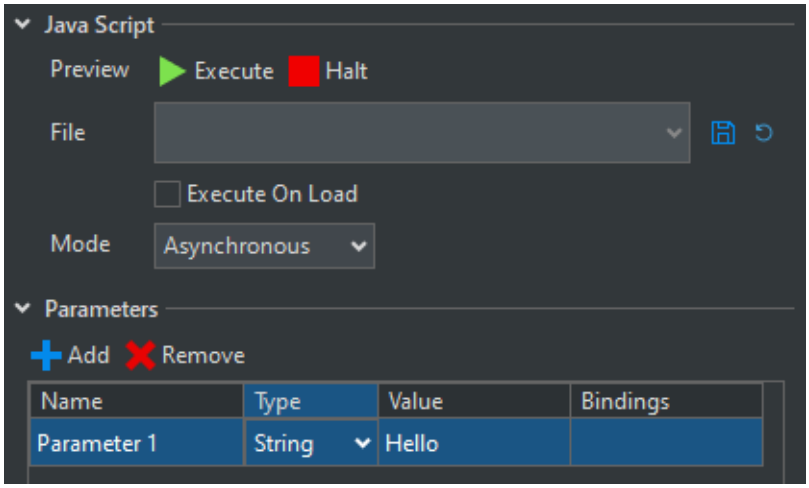
- Java Script Effect can be applied to any scene Graphic Object.
- Java Script Resource can be applied to the Scene, Project, and Application levels as well as Master Control Panels.

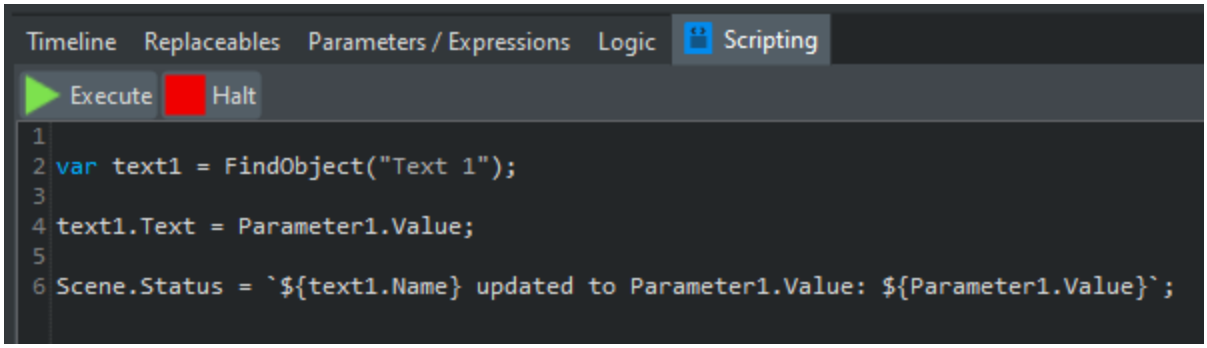


To edit a JavaScript: click the edit pencil icon. This will open the Scripting panel within Prime.



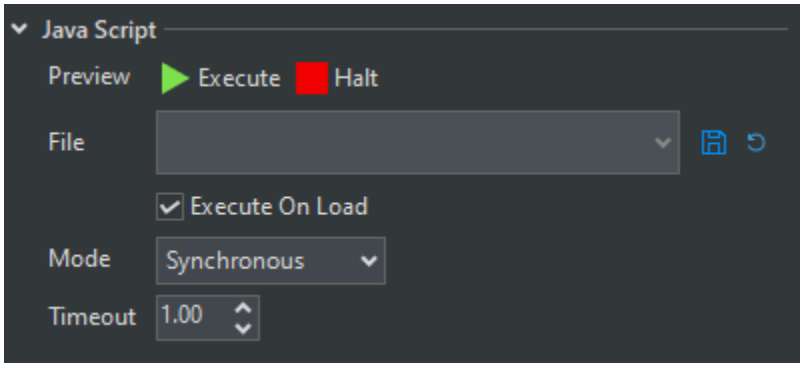
From here, you can set options, add parameters, edit the code in the Scripting window, and Execute or Halt scripts for testing.





Execute On Load

Checking this option will cause the script to execute automatically when the scene is loaded in playout



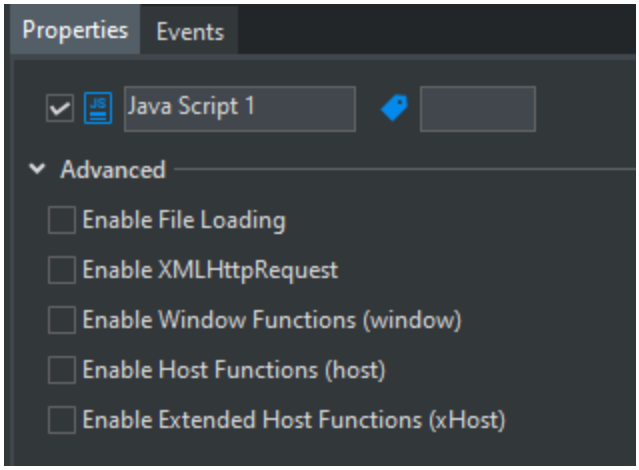
Mode

Allows changing whether the script is run in Asynchronous mode (on a background thread), or in Synchronous mode (blocking until finished). Changing to Synchronous Mode will enable the Timeout property

Timeout

Specifies the number of seconds that will run in Synchronous mode before timing out.

JavaScript Advanced Options



Enable File Loading

Allows code defined in external .js files to be imported into the script using the import keyword. The default location of module files is the Scripts folder of the Project folder belonging to the current project.

Enable XMLHttpRequest

Allows creating a new XMLHttpRequest object that can be used for downloading data from a server.

Enable Window Functions

Enables window functions within the script. For example alert, confirm, prompt, MsgBox and InputBox.

Enable Host functions

Enables the host keyword within the script.
See the HostFunctions section below for more info.

Enable Extended Host functions

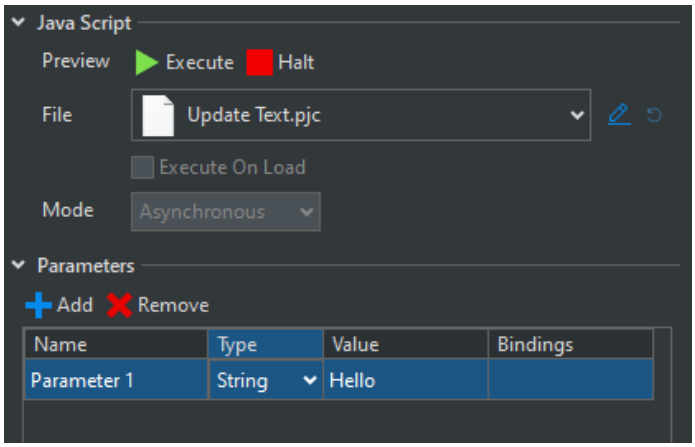
Enables the xHost keyword within the script. See the ExtendedHostFunctions section below for more info.

Java Script Parameters

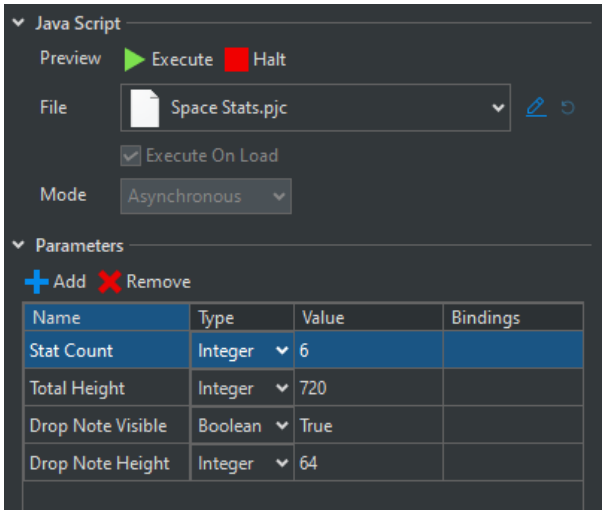
Click Add and Remove to edit the parameters of the selected JavaScript object. Any Parameters added to the JavaScript effect will be saved with the effect. All child parameter objects are natively accessible from within the script code.

To Save a new script:

Click the disk icon and select Save or Save As.



To import an existing JavaScript: Navigate to script file using the File browser



Keywords

The following keywords are available to use in all JavaScript objects running within the application

Keywords	Type	Description
Scene	Scene	Represents the scene object that the script is apart of
Parent	SceneObject	Represents the parent object to which the JavaScript object belongs
This	JavaScript	Represents the actual JavaScript object itself
Project	Project	Represents the project that the scene is apart of
Channel	Channel	Represents the channel that the scene is loaded on
Playout	Playout	Represents an object that can access all channels and clip players that are configured
window	Window	Represents an object that can be used to invoke window functions (alert, confirm, prompt, MsgBox and InputBox). Note: Enable Window Functions must be checked in the advanced section of the script
host	HostFunctions	Represents an object that can be used to invoke host functions within the script. Note: Enable Host Functions must be checked in the advanced section of the script
xHost	ExtendedHost Functions	Represents an object that can be used to invoke extended host functions within the script. Note: Enable Extended Host Functions must be checked in the advanced section of the script
FindObject(string name)	SceneObject	Returns the nearest object to the parent JavaScript that matches the name argument

All parameters defined in the JavaScript object are also natively available as keywords to be used within the script.

Properties

The following types are common to object properties

Size

Name	Type	Description
Width	double	Gets or sets Width value of the vector
Height	double	Gets or sets Height value of the vector

Point

Name	Type	Description
X	double	Gets or sets X value of the point
Y	double	Gets or sets Y value of the point

Vector

Name	Type	Description
X	double	Gets or sets X value of the vector
Y	double	Gets or sets Y value of the vector
Z	double	Gets or sets Z value of the vector
Link	bool	Gets or sets a value that links the axis values together proportionally. This is only used for the Scale property

Color

Name	Type	Description
R	byte	Gets the red value of the color
G	byte	Gets the green value of the color
B	byte	Gets the blue value of the color
A	byte	Gets the alpha value of the color

Note: colors can be assigned by setting a hex code string: text1.Color = "#FF0000"

Font

Name	Type	Description
Name	string	Gets or sets the name of the font
Size	double	Gets or sets size of the font
Color	Color	Gets or sets the color of the text object
Kerning	double	Gets or sets the kerning of the text object
Leading	double	Gets or sets the leading of the text object
SpaceWidth	double	Gets or sets the space width of the text object
FixedPitch	double	Gets or sets the fixed pitch of the text object

Outline

Name	Type	Description
Enabled	bool	Gets or sets the enabled property of the outline
Color	Color	Gets or sets the color property of the outline
Size	int	Gets or sets the size of the outline

Shadow

Name	Type	Description
Enabled	bool	Gets or sets the enabled property of the shadow
Color	Color	Gets or sets the color property of the shadow
Blur	int	Gets or sets the blur of the shadow
Offset	Vector	Gets a Vector object that represents the offset of the shadow

TimeCode

Name	Type	Description
Interlaced	bool	Gets a value that represents the interlaced value of the timecode
Rate	double	Gets the rate of the timecode (25, 29.97, 50, 59.94, …)
TotalFrames	int	Gets the total number of frames represented by the timecode
TotalFields	int	Gets the total number of fields represented by the timecode. This value will be the same as TotalFrames if the timecode is not interlaced
TotalSeconds	double	Gets the total number of seconds represented by the timecode
Fields	int	Gets the number of truncated fields represented by the timecode. This value will be the same as Frames if the timecode is not interlaced

Frames	int	Gets the number of truncated frames represented by the timecode
Seconds	int	Gets the number of truncated seconds represented by the timecode
Minutes	int	Gets the number of truncated minutes represented by the timecode
Hours	int	Gets the number of hours represented by the timecode
Text	string	Gets a string representation of the timecode in "hh:mm:ss.ff" format

Resolution

Name	Type	Description
Width	double	Gets or sets Width value of the resolution
Height	double	Gets or sets Height value of the resolution
FrameRate	FrameRate	Gets an object that represents the Frame Rate of the resolution
Interlaced	bool	Gets a value that represents the interlaced value of the resolution

Bindings

Name	Type	Description
Count	int	Gets the bindings count
Text	string	Gets or sets the text representation of the binding. Multiple bindings are separated by semicolons
Add(string binding)	void	Adds a binding to the object
Remove(string binding)	void	Removes a binding from the object
Clear()	void	Clears all bindings from the object
GetTargetObject()	SceneObject	Gets the first object referenced in the bindings
GetTargetProperty()	string	Gets the the property of the first object referenced in the bindings
GetTargetValue()	object	Gets the value of the property of the first object referenced in the bindings

Objects

SceneObject

The following properties and methods are common to all objects within the scene tree

Name	Type	Description
Name	string	Gets or sets the name of the scene
Type	string	Gets the type of the object (Text, Image, Clip,...)
Tag	string	Gets or sets the Tag property which can be used mark objects to be designated for custom logic
Parent	SceneObject	Gets the parent of the object
GetValue(string property)	object	Returns the value of the specified property
SetValue(string property, object value)	void	Sets the specified property to the value argument
FindObject(string name)	SceneObject	Returns the nearest object that matches the name argument

SceneObjectList

Represents a list of child objects

Name	Type	Description
Count	int	Gets the count of objects in the list
New(string type)	SceneObject	Creates and returns a new SceneObject based on the provided type ("Text", "Image", "Clip", "Group"...) Note this object is not automatically added to the list
Add(SceneObject sceneObject)	void	Adds the specified object to the end of the list
Insert(int index, SceneObject sceneObject)	void	Inserts the specified object at the specified index location
Remove(SceneObject sceneObject)	bool	Removes the specified object from the list. Returns value based on success

Scene

Scene include all fields from SceneObject and the following fields

Name	Type	Description
Description	string	Gets or sets the scene description
Keywords	string	Gets or sets the keywords for the scene
Style	string	Gets or sets the style of the scene
Layer	int	Gets or sets the layer of the scene
EffectIn	TriggerList	Gets the trigger list object that represents Effect In
EffectOut	TriggerList	Gets the trigger list object that represents Effect Out
LayerIn	TriggerList	Gets the trigger list object that represents Layer In
LayerOut	TriggerList	Gets the trigger list object that represents Layer Out
PreviewIn	TriggerList	Gets the trigger list object that represents Preview In
Resolution	Resolution	Gets the resolution of the scene
FrameRate	FrameRate	Gets the frame rate of the scene
FilePath	string	Gets the full path of the scene
Directory	string	Gets the directory that contains the scene file
Active	bool	Gets or sets value that indicates if the scene is the active scene of the channel
Status	string	Sets a status message that will be displayed in the status bar of the scene in the playout panel
Loaded	bool	Gets a value that indicates if the scene is loaded
Playing	bool	Gets a value that indicates if the scene is playing
Updating	bool	Gets a value that indicates if the scene is updated by a matching scene from preview
Closed	bool	Gets a value that indicates if the scene is closed
Project	Project	Gets the project that contains the scene
Canvas	Canvas	Gets the canvas object used for the display of graphics and effects.
Resources	SceneResources	Gets the list of resources defined for this scene such as DataObjects,
Parameters	ParameterList	Gets the list of parameters defined for the scene.
Expressions	ExpressionList	Gets the list of expressions defined for the scene.
Conditions	ConditionList	
Replaceables	ReplaceableList	
Actions	ActionList	Gets the list of Actions defined for this scene.
AllObjects	Enumerable<SceneObject>	Gets the list of all objects including graphics, controls, resources and effects that the scene contains
Play()	void	Plays the scene
Stop()	void	Stops the scene
Close()	void	Closes the scene
Update(string name, string value)	bool	Attempts to update the default value of an object matching the name argument with the value

		arguments. Returns true if successful, and false otherwise
GetObjectValue(string name, string property)	object	Gets the value of the property argument that matches an object in the scene that matches the name argument
SetObjectValue(string name, string property, object value)	void	Sets the value of a the property argument that matches an object in the scene that matches the name argument
Execute(string command)	void	Executes a command as if it was a logic statement. For instance: Scene.Execute("Image1.Opacity = 100")
Evaluate(string expression)	object	Evaluates an expression as if it was in a logic statement and returns the result. For instance: Scene.Evaluate("Text1.Text")

QueueUpdates()	QueueCommand	Puts the scene into Queue Mode and returns a QueueCommand that can be used to apply all queued updates simultaneously. For example:
		
		var updates = Scene.QueueUpdates();
		
		try
		{
		  // update multiple objects in scene
		}
		finally
		{
		  updates.Execute();
		}

SaveMessage()	string	Saves the current state of the scene to a message file in the Messages folder of the Project of the scene. The first available message number will be used as the name of the message. The path of the message file will be returned.
SaveMessage(string message)	string	Saves the current state of the scene to a message file in the Messages folder of the Project of the scene. The message argument can be the name of the message or the message path. The path of the message file will be returned.
LoadedChanged	SceneEvent	Represents an event that can be connected to a function to be run every time the Loaded property changes. For example Scene.LoadedChanged.connect(OnLoadedChanged); function OnLoadedChanged(scene)

		{
		// react to loaded changed
		}

PlayingChanged	SceneEvent	Represents an event that can be connected to a function to be run every time the Playing property changes.
UpdatingChanged	SceneEvent	Represents an event that can be connected to a function to be run every time the Updated property changes.
StyleChanged	SceneEvent	Represents an event that can be connected to a function to be run every time the Style property changes.
AfterLoad	SceneEvent	Represents an event that can be connected to a function to be run after the scene has finished loading.
BeforePlay	SceneEvent	Represents an event that can be connected to a function to be run before the scene has started playing.
AfterPlay	SceneEvent	Represents an event that can be connected to a function to be run after the scene has finished playing.
BeforeUpdate	SceneEvent	Represents an event that can be connected to a function to be run before the scene has started updating replaceable values.
AfterUpdate	SceneEvent	Represents an event that can be connected to a function to be run after the scene has finished updating replaceable values.
BeforeStop	SceneEvent	Represents an event that can be connected to a function to be run before the scene has started stopping.
AfterStop	SceneEvent	Represents an event that can be connected to a function to be run after the scene has finished stopping.
BeforeClose	SceneEvent	Represents an event that can be connected to a function to be run before the scene has started closing.

Action

Represents a collection of animations

Name	Type	Description
Name	string	Gets or sets the name of the action
Length	TimeCode	Gets an object representing the length of the action
Loop	bool	Gets or sets the loop property for all of the associated animations
Animations	Enumerable<Animation>	Gets an object that can be used to iterate over all associated animations
Play()	void	Plays the animation
Play(int offset)	void	Plays the animation after the specified offset
Stop()	void	Stops the animation

Animation

Represents a child animation of an object

Name	Type	Description
Play()	void	Plays the animation
Play(int offset)	void	Plays the animation after the specified offset
Stop()	void	Stops the animation
Animate(SceneObject sceneObject)	void	Plays the animation on the provided object
Animate(SceneObject sceneObject, int offset)	void	Plays the animation on the provided object after the specified offset

AnimationList

Represents the list of child animations of an object

Name	Type	Description
Default	Animation	Returns the Default animation
Find(string name)	Animation	Returns the animation specified by the provided name if it exists, or null if not

DynamicObject

DynamicObject includes all fields from SceneObject and the following fields

Name	Type	Description
DefaultAnimation	Animation	Gets or sets the Default animation
Animations	AnimationList	Gets an object representing the animations defined

Graphic

Graphic includes all fields from DynamicObject and the following fields

Name	Type	Description
Enabled	bool	Gets or sets whether the graphic is visible or not
Opacity	double	Gets or sets the opacity of the graphic
Position	Vector	Gets a vector object that represents the Position of the graphic
Scale	Vector	Gets a vector object that represents the Scale of the graphic
Rotation	Vector	Gets a vector object that represents the Rotation of the graphic
Pivot	Vector	Gets a vector object that represents the Pivot of the graphic
Effects	SceneObject List<Effect>	Gets an object that represents the child effects of the graphic

2D Graphic

2D Graphic includes all fields from Graphic and the following fields

Name	Type	Description
Width	double	Gets or sets the width of the graphic
Height	double	Gets or sets the height of the graphic
Size	Size	Gets a size object that represents the size of the graphic
Origin	Point	Gets a point object that represents the origin of the graphic

Text

Text objects include all fields from 2D Graphic and the following fields

Name	Type	Description
Text	string	Gets or sets the Text of the text object
HorizontalAlignment	HorizontalAlignment	Gets or sets the horizontal alignment of the text object: "Left", "Center", "Right", "FirstCharacter", "LastCharacter"
HorizontalAlignmentCharacter	string	
VerticalAlignment	VerticalAlignment	Gets or sets the vertical alignment of the text object: "Top", "First", "Middle", "Last", "Bottom"
WordWrap	WordWrap	Gets or sets the word wrap of the text object: "Off", "Wrap" or "UniformScale"
HorizontalScale	HorizontalScale	Gets or sets the horizontal scale of the text object: "Off", "ScaleToFit" or "KernToFit"
VerticalScale	VerticalScale	Gets or sets the vertical scale of the text object: "Off", "ScaleToFit" or "FillToFit"
AllCaps	bool	Gets or sets the all caps property of the text object
CapsRatio	double	Gets or sets the caps ratio of the text object
Numeric	bool	Gets or sets the numeric property of the text object
Number	double	Gets or sets the number property of the text object
Format	string	Gets or sets the format property of the text object when Numeric is enabled
Font	Font	Gets an object that represents the font text object containing the following properties:
• **Name**	string	Gets or sets the name of the font
• **Size**	double	Gets or sets size of the font
• **Color**	Color	Gets or sets the color of the text object
• **Kerning**	double	Gets or sets the kerning of the font
• **Leading**	double	Gets or sets the leading of the text object
• **SpaceWidth**	double	Gets or sets the space width of the text object
• **FixedPitch**	double	Gets or sets the fixed pitch of the text object
TextMode	TextMode	Gets or sets the mode of the text object ("_2d" or "_3d")
_2d	_2dTextProperties	Gets an object that represents the 2D text properties containing the following properties:

• **Resolution**	double	Gets or sets resolution of the 2D font
• **LockResolution**	bool	Gets or sets a value that locks the resolution to the font size
• **ColorMode**	ColorMode	Gets or sets the color mode of the 2D font ("Solid", "Linear", "Quad" or "DoubleVertical")
• **GradientTarget**	GradientTargetMode	Gets or sets the gradient target of the 2D font used for non Solid color modes ("Character", "Row" or "Object")
• **StartColor**	Color	Gets or sets the linear start color used for the Linear color mode
• **FinishColor**	Color	Gets or sets the linear finish color used for the Linear color mode
• **GradientAngle**	double	Gets or sets the angle direction of start and finish colors when ColorMode is Linear
• **GradientColor1**	Color	Gets or sets the first color used when ColorMode is set to Quad or DoubleVertical
• **GradientColor2**	Color	Gets or sets the second color used when ColorMode is set to Quad or DoubleVertical
• **GradientColor3**	Color	Gets or sets the third color used when ColorMode is set to Quad or DoubleVertical
• **GradientColor4**	Color	Gets or sets the fourth color used when ColorMode is set to Quad or DoubleVertical
• **Outline**	Outline	Gets an object that represents the Outline edge of the text object
• **Shadow**	Shadow	Gets an object that represents the Shadow edge of the text object
Style	string	Gets or sets a file that represents the Mode, Font, 2D and 3D properties of the text object
LinkStyle	bool	Gets or sets a value that represents whether the text object style is linked to the file specified in the Style property
LineCount	int	Gets the line count of the text object
UpdateTextSize	bool	Gets or sets a value that enables the TextSize property to be updated with text size as the text changes
TextSize	TextSize	Gets a object that represents
GetTextBounds(string text)	Rectangle	Returns a rectangle object that represents the bounds that the provided text would occupy
TextChanged	SceneObjectEvent	Represents an event that can be connected to a function to be run every time the Text property changes.

		For example
		
		var text = FindObject("Text 1");
		
		text.TextChanged.connect(OnTextChanged);
		
		function OnTextChanged(textObject)
		{
		  // react to text changed
		}

Image

Image objects include all fields from 2D Graphic and the following fields

Name	Type	Description
File	string	Gets or sets the file of the image object
HideOnClear	bool	Gets or sets a value that causes the image to disappear when the File property set to an empty string
ColorMode	ColorMode	Gets or sets the color mode of the image ("Solid", "Linear", "Quad" or "DoubleVertical")
Color	Color	Gets or sets the color of the image object
StartColor	Color	Gets or sets the linear start color used for the Linear color mode
FinishColor	Color	Gets or sets the linear finish color used for the Linear color mode
GradientAngle	double	Gets or sets the angle direction of start and finish colors when ColorMode is Linear
GradientColor1	Color	Gets or sets the first color used when ColorMode is set to Quad or DoubleVertical
GradientColor2	Color	Gets or sets the second color used when ColorMode is set to Quad or DoubleVertical
GradientColor3	Color	Gets or sets the third color used when ColorMode is set to Quad or DoubleVertical
GradientColor4	Color	Gets or sets the fourth color used when ColorMode is set to Quad or DoubleVertical
Shadow	Shadow	Gets an object that represents the Shadow edge of the imageobject
FileChanged	SceneObjectEvent	Represents an event that can connected to a function to be run every time the File property changes

Clip

Clip objects include all fields from 2D Graphic and the following fields

Name	Type	Description
File	string	Gets or sets the file of the clip
Color	Color	Gets or sets the color of the clip
Loop	bool	Gets or sets the loop value of the clip
LoopCount	int	Gets or sets the count for looping
Speed	double	Gets or sets the speed of the clip
Volume	double	Gets or sets the volume of the clip
Muted	bool	Gets or sets whether the clip is muted
HoldLastFrame	bool	Gets or sets a value that controls the behavior that occurs after the clip is finished: the last frame is held if true, or clip goes if false
Length	TimeCode	Gets an object representing the length of the clip
CurrentFrame	TimeCode	Gets an object representing the current frame of the clip
Cue()	void	Sets the clip to the first frame
Play()	void	Plays the clip from the beginning
Stop()	void	Stops and hides the clip
Pause()	void	Pauses the clip at the current frame
Resume()	void	Resumes the clip from the current frame
End()	void	Sets the clip to the last frame
Jog(int frames)	void	Offsets the clip by the specified number of frames
Mute()	void	Mutes the clip
Unmute()	void	Unmutes the clip
FileChanged	SceneObjectEvent	Represents an event that can connected to a function to be run every time the File property changes

Group

Group objects include all fields from Graphic and the following fields

Name	Type	Description
Graphics	SceneObjectList<Graphic>	Gets an object representing the list of child graphics

ObjectsRecursive	Enumerable<SceneObject>	Gets an object that can be used to iterate over all children objects recursively. This will include both graphics and effects. For example:
		
		var sceneGroup = Scene.SceneGroup;
		
		for (var object in sceneGroup.ObjectsRecursive)
		{
		  // operate on object
		}

Control

Controls include all fields from Scene Object and the following fields

Name	Type	Description
Enabled	bool	Gets or sets the enabled property of the control
Visible	bool	Gets or sets the visible property of the control
X	int	Gets or sets the X position of the control
Y	int	Gets or sets the Y position of the control
Width	int	Gets or sets the width of the control
Height	int	Gets or sets the height of the control
Location	Point	Gets the location of the control
Size	Size	Gets the size of the control
BackColor	Color	Gets the back color of the control
ForeColor	Color	
Text	string	Gets the text of the control
Bindings	Bindings	Gets the bindings of the control
GetTargetObject()	SceneObject	Gets the first object referenced in the bindings
GetTargetProperty()	string	Gets the the property of the first object referenced in the bindings
GetTargetValue()	object	Gets the value of the property of the first object referenced in the bindings
TextChanged	SceneObjectEvent	Represents an event that can be connected to a function to be run every time the Text property changes

Control Panel

Control Panels include all fields from Control and the following fields

Name	Type	Description
ActiveControl	Control	Gets or sets the active control of the Control Panel
Controls	SceneObjectList<Controls>	Gets an object representing the list of child controls

Button

Buttons include all fields from Control and the following fields

Name	Type	Description

Click	SceneObjectEvent	Represents an event that can be connected to a function to be run every time the button is clicked. For example
		
		var button = FindObject("Button 1");
		
		button.Click.connect(OnButtonClicked);
		
		function OnButtonClicked(buttonObject)
		{
		  // react to button clicked
		}

Check Box

Check boxes include all fields from Control and the following fields

Name	Type	Description
Checked	bool	Gets or sets the Checked property of the text box
CheckedChanged	SceneObjectEvent	Represents an event that can be connected to a function to be run every time the Checked property changes

Text Box

Text boxes include all fields from Control and the following fields

Name	Type	Description
Multiline	bool	Gets or sets the Multiline property of the text box
SelectedText	string	Gets or sets the selected text of the text box
SelectionStart	int	Gets or sets the beginning index of the text box selection
SelectionLength	int	Gets or sets the length of the text box selection

Combo Box

Combo boxes include all fields from Control and the following fields

Name	Type	Description
Items	ComboBoxItems	Gets an object representing combo box items

Parameter

Includes all fields from SceneObject plus the following fields

Name	Type	Description
Enabled	bool	Gets or sets a value representing the enabled property of the parameter
Value	object	Gets or sets the value of the parameter
PreviousValue	object	Gets the previous value of the parameter
Bindings	Bindings	Gets the bindings for the parameter

ValueChanged	SceneObjectEvent	Represents an event that can be connected to a function to be run every time the Value property changes. For example
		
		var text = FindObject("Text 1");
		
		Parameter1.ValueChanged.connect(OnValueChanged);
		
		function OnValueChanged(parameter)
		{
		  // react to parameter value changed
		}

JavaScript

Includes all fields from SceneObject plus the following fields

Name	Type	Description
Sync	bool	Gets or sets the Sync property of the script that determines whether the script is run synchronously or asynchronously. For example, to enable: This.Sync = true;
Timeout	double	Gets or sets the timeout in seconds that is used to wait for synchronous scripts to finish. The default value is 1 second
Parameters	ParametersList	Gets an object representing the list of child parameters

Project

Includes all fields from Scene plus the following fields

Name	Type	Description
Path	string	Gets the directory path of the Project

Channel

Represents an object that can be used to affect the scenes loaded and played in a channel

Name	Type	Description
Index	int	Gets the 0 based index of the channel as defined by the order in the playout configuration
Number	int	Gets the 1 based index of the channel as defined by the order in the playout configuration
Name	string	Gets the name of the channel as defined by the order in the playout configuration
Resolution	Size	Gets the resolution of the channel as defined by the playout configuration
ActivePreviewScene	Scene	Gets the active scene in preview
ActiveProgramScene	Scene	Gets the active scene in program
OpenScenes	Enumerable<Scene>	Gets an object that can be used to iterate over all of the open scenes in the channel

GetScene(string name)	Scene	Returns the scene in the channel that matches the name argument if it exists, or null if not
LoadScene(string name)	bool	Loads the specified scene name into the channel. Returns the success of finding the scene. Note: the name can be a file name within the current project, or a full file path
PlayScene(string name)	bool	Plays the specified scene name on the channel. Returns the success of finding the scene
StopScene(string name)	bool	Stops the specified scene name on the channel. Returns the success of finding the scene
CloseScene(string name)	bool	Closes the specified scene name on the channel. Returns the success of finding the scene
RenderImage(string path = null)	string	Renders a scene to an image file. If the provided path is an image file extension, then the playing scene(s) on the Render channel will be rendered to this image path. If the provided path is a scene or message extension, then this asset will be rendered to a temporary image path. If a path is not provided, then the playing scene(s) on the Render channel will be rendered to a temporary image path. Returns the location of the image file
RenderImage(string scenePath, string imagePath)	string	Renders a scene to an image file. Renders the provided scenePath to the provided imagePath. Returns the imagePath
RenderClip(string scenePath, clipPath = null, object parameters = null)	string	Renders a scene to a clip file. If the scenePath is empty, then the playing scene on the Render channel will be rendered. Otherwise, the scenePath asset will be used as the scene to be rendered. If the clipPath is empty, then a temporary path will be assigned. Otherwise, this

		argument will be the output location of the rendered clip. An optional parameters argument can be provided for additional control over the rendered clip. The argument can either be JSON text or a JavaScript object with any of the following properties: ● Size - parent object with integer Width and Height child properties to specify the output size of the clip ● TotalFields - integer property used to specify the number of fields the output clip should contain ● CropRectangle - parent object with integer Left, Top, Right and Bottom child properties to specify the number of pixels to crop from each side ● EnforceEffectOut - bool property used to specify whether the Effect Out of the scene should be rendered ● Timeout - integer property used to specify the number of milliseconds allowed for the render before timing out ● Encoding - string property used to set the encoding of the clip: ○ Gtc, ○ GtcJpeg, ○ GtcLzo, ○ QuicktimeRle, ○ MotionJpeg, ○ DnxHdHq, ○ DnxHdSq, ○ DnxHdLb, ○ DnxHrHq, ○ DnxHrHqAlpha, ○ DnxHrSq, ○ DnxHrSqAlpha, ○ DnxHrLb, ○ DnxHrLbAlpha, ○ Vp9

		Returns the location of the rendered clip file

ClipPlayer

Represents an object that can be used to affect the clips cued and played in a clip player

Name	Type	Description
Index	int	Gets the 0 based index of the clip player as defined by the order in the playout configuration
Number	int	Gets the 1 based index of the clip player as defined by the order in the playout configuration
Name	string	Gets the name of the clip player as defined by the order in the playout configuration
CuedClip	Scene	Gets the cued clip scene
PlayingClip	Scene	Gets the playing clip scene
ClipExists(string name)	bool	Returns true if the specified clip name exists within the project, false if not
CueClip(string name)	bool	Loads the specified scene file into the channel. Returns the success of finding the scene. Note: the file can be a file name within the current project, or a full file path
PlayCued()	void	Plays the current cued clip
ClearCued()	void	Clears the current cued clip
PausePlaying()	void	Pauses the playing clip
ResumePlaying()	void	Resumes the playing clip
StopPlaying()	void	Stops the playing clip
ClearPlaying()	void	Clears the playing clip
JogPlaying(int frames)	void	Jogs the playing clip by the specified number of frames

Playout

Represents an object that can access all channels and clip players that are configured

Name	Type	Description
Channels	ReadOnlyList<Channel>	Gets a read only list that represents the channels defined in the current playout configuration
ClipPlayers	ReadOnlyList<ClipPlayer>	Gets a read only list that represents the clip players defined in the current playout configuration
ChannelExists(int number)	bool	Returns true if a channel exists at this number
ChannelExists(string name)	bool	Returns true if a channel exists at this name
GetChannel(int number)	Channel	Returns the channel object at the specified number
GetChannel(string name)	Channel	Returns the channel object at the specified name
ClipPlayerExists(int number)	bool	Returns true if a clip player exists at this number
ClipPlayerExists(string name)	bool	Returns true if a clip player exists at this name
GetClipPlayer(int number)	ClipPlayer	Returns the clip player object at the specified number
GetClipPlayer(string name)	ClipPlayer	Returns the clip player object at the specified name

LastAutomationCommand	LastAutomationCommand	Returns the Text and Time of the most recently received Intelligent Interface command.
		
		var text = Playout.LastAutomationCommand.Text; //return the Text of the Command
		
		var time = Playout.LastAutomationCommand.Time; //returns the data/time of when command was received

Window

Represents an object that can be used to display popup message boxes

Name	Type	Description
alert(string message)	void	Displays an OK message box with the provided message text https://developer.mozilla.org/en-US/docs/Web/API/Window/alert
confirm(string message)	bool	Displays an OK/Cancel message box with the provided message text. Returns true if OK is clicked and false if Cancel is clicked https://developer.mozilla.org/en-US/docs/Web/API/Window/confirm
prompt(string message, string defaultValue = "")	string	Displays an OK/Cancel input box with the provided message text and optional default input value. Returns the input text if OK is clicked, or blank if Cancel is clicked https://developer.mozilla.org/en-US/docs/Web/API/Window/prompt
MsgBox(string message)	void	Displays an OK message box with the provided message text
InputBox(string message, string title = "", string defaultValue = "", int x = -1, int y = -1)	string	Displays an OK/Cancel input box with the provided message text, optional title and optional default input value. Returns the input text if OK is clicked, or blank if Cancel is clicked

HostFunctions

The HostFunctions object provides access to basic .NET functionality as defined here: https://microsoft.github.io/ClearScript/Reference/html/T_Microsoft_ClearScript_HostFunctions.htm

For example, a .NET array can be created if necessary:

```
var array = host.newArr(10);
```

ExtendedHostFunctions

The ExtendedHostFunctions object provides access to extended .NET functionality as defined here: https://microsoft.github.io/ClearScript/Reference/html/T_Microsoft_ClearScript_HostFunctions.htm

For example, the .NET File class can be referenced to access file information on disk:

```
var File = xHost.type("System.IO.File");

var text = File.ReadAllText(filePath);
```

Other libraries can be accessed for specific classes:

```
var DrawingLib = xHost.lib("System.Drawing");

var Color = DrawingLib.System.Drawing.Color;
```

ABOUT US

Chyron is ushering in the next generation of storytelling in the digital age. Founded in 1966, the company pioneered broadcast titling and graphics systems. With a strong foundation built on over 50 years of innovation and efficiency, the name Chyron is synonymous with broadcast graphics. Chyron continues that legacy as a global leader focused on customer-centric broadcast solutions. Today, the company offers production professionals the industry's most comprehensive software portfolio for designing, sharing, and playing live graphics to air with ease. Chyron products are increasingly deployed to empower OTA & OTT workflows and deliver richer, more immersive experiences for audiences and sports fans in the arena, at home, or on the go.

CONTACT SALES

EMEA • North America • Latin America • Asia/Pacific
+1.631.845.2000 • sales@chyron.com

Chyron