GEMS API Reference

Travis L. Seymour, PhD 2026

This document describes all conditions, triggers, and actions available in GEMS (Graphical Environment Management System).

Restriction Scopes

Actions, conditions, and triggers can be restricted to specific contexts:

Scope	Description
`view`	Can be used at the view level
`object`	Can be used on objects within a view
`global`	Can be used in global scripts
`pocket`	Can be used on pocket objects

Triggers

Triggers define events that initiate script execution.

ViewTimePassed

```
ViewTimePassed(Seconds: float)
```

Description: Fires when the specified number of seconds has elapsed since entering the current view.

Parameters:

- `Seconds` (float): Time in seconds to wait before triggering

Restrictions: view

TotalTimePassed

```
TotalTimePassed(Seconds: float)
```

Description: Fires when the specified number of seconds has elapsed since the environment started running.

Parameters:

- `Seconds` (float): Total elapsed time in seconds

Restrictions: global

MouseClick

```
MouseClick()
```

Description: Fires when the user clicks on an object or pocket.

Parameters: None

Restrictions: object, pocket

NavLeft

```
NavLeft()
```

Description: Fires when the user clicks in the left navigation area of the view.

Parameters: None

Restrictions: view

NavRight

```
NavRight()
```

Description: Fires when the user clicks in the right navigation area of the view.

Parameters: None

Restrictions: view

NavTop

```
NavTop()
```

Description: Fires when the user clicks in the top navigation area of the view.

Parameters: None

Restrictions: view

NavBottom

```
NavBottom()
```

Description: Fires when the user clicks in the bottom navigation area of the view.

Parameters: None

Restrictions: view

DroppedOn

```
DroppedOn(Object: objnum)
```

Description: Fires when a specific object is dropped onto this object or pocket.

Parameters:

- `Object` (objnum): The ID of the object that must be dropped

Restrictions: object, pocket

KeyPress

```
KeyPress(Key: key)
```

Description: Fires when the specified key is pressed.

Parameters:

- `Key` (key): The key to listen for

Restrictions: view, global

Conditions

Conditions are evaluated to determine whether associated actions should execute.

VarValueIs

```
VarValueIs(Variable: varname, Value: value)
```

Description: Returns true if the specified variable exists and equals the given value.

Parameters:

- `Variable` (varname): Name of the variable to check
- `Value` (value): Value to compare against

Restrictions: view, object, global, pocket

VarValueIsNot

```
VarValueIsNot(Variable: varname, Value: value)
```

Description: Returns true if the specified variable does not equal the given value (or doesn’t exist).

Parameters:

- `Variable` (varname): Name of the variable to check
- `Value` (value): Value to compare against

Restrictions: view, object, global, pocket

VarExists

`VarExists(Variable: varname)`

Description: Returns true if the specified variable exists.

Parameters:

- `Variable` (varname): Name of the variable to check

Restrictions: view, object, global, pocket

ViewTimePassed

`ViewTimePassed(Seconds: number)`

Description: Returns true if the specified time has passed since entering the current view.

Parameters:

- `Seconds` (number): Time in seconds

Restrictions: view, object

TotalTimePassed

`TotalTimePassed(Seconds: number)`

Description: Returns true if the specified time has passed since the environment started.

Parameters:

- `Seconds` (number): Time in seconds

Restrictions: view, object, global, pocket

VarCountEq

```
VarCountEq(Count: number)
```

Description: Returns true if the total number of variables equals the specified count.

Parameters:

- `Count` (number): Expected variable count

Restrictions: view, object, global, pocket

VarCountGtEq

```
VarCountGtEq(Count: number)
```

Description: Returns true if the total number of variables is greater than or equal to the specified count.

Parameters:

- `Count` (number): Minimum variable count

Restrictions: view, object, global, pocket

VarCountLtEq

```
VarCountLtEq(Count: number)
```

Description: Returns true if the total number of variables is less than or equal to the specified count.

Parameters:

- `Count` (number): Maximum variable count

Restrictions: view, object, global, pocket

KeyBufferContains

```
KeyBufferContains(Keys: value, IgnoreCase: bool)
```

Description: Returns true if the key buffer contains the specified key sequence.

Parameters:

- `Keys` (value): Key sequence to search for
- `IgnoreCase` (bool): If true, comparison is case-insensitive

Restrictions: view, global

Actions

Actions are commands that modify the environment state or trigger effects.

PortalTo

```
PortalTo(View: viewnum, VidFile: vidfile = "")
```

Description: Navigates to the specified view. If a video file is provided, plays it as a fullscreen transition before changing views. Right-clicking the video skips to the destination view immediately.

Parameters:

- `View` (viewnum): ID of the destination view
- `VidFile` (vidfile): Optional video file for transition effect

Restrictions: view, object, global, pocket

PlaySound

```
PlaySound(SoundFile: sndfile, Asynchronous: bool, Volume: 01float,
Loop: bool)
```

Description: Plays the specified audio file. Asynchronous playback returns control immediately; synchronous playback blocks until complete.

Parameters:

- `SoundFile` (sndfile): Path to the audio file
- `Asynchronous` (bool): If true, plays without blocking
- `Volume` (01float): Volume level (0.0 to 1.0)

- `Loop` (bool): If true, loops the audio continuously

Restrictions: view, object, global, pocket

PlayBackgroundMusic

```
PlayBackgroundMusic(SoundFile: sndfile, Volume: 01float, Loop:
bool)
```

Description: Plays the specified audio file as background music. Only one background music stream can play at a time - calling this while music is playing stops the current music first. Background music persists across view changes and is not affected by StopAllSounds.

Parameters:

- `SoundFile` (sndfile): Path to the audio file
- `Volume` (01float): Volume level (0.0 to 1.0)
- `Loop` (bool): If true, loops the music indefinitely

Restrictions: view, object, global, pocket

StopBackgroundMusic

```
StopBackgroundMusic()
```

Description: Stops the currently playing background music.

Parameters: None

Restrictions: view, object, global, pocket

StopSound

```
StopSound(SoundFile: sndfile)
```

Description: Stops playback of the specified sound file if currently playing.

Parameters:

- `SoundFile` (sndfile): Path to the audio file to stop

Restrictions: view, object, global, pocket

StopAllSounds

```
StopAllSounds()
```

Description: Stops all currently playing sound effects. Does not affect background music.

Parameters: None

Restrictions: view, object, global, pocket

PlayVideo

```
PlayVideo(VidFile: vidfile, Start: number, Left: number, Top:
number, Volume: 01float, Loop: bool)
```

Description: Plays a video file at the specified position. Right-click to close the video.

Parameters:

- `VidFile` (vidfile): Path to the video file
- `Start` (number): Start time in seconds
- `Left` (number): X position
- `Top` (number): Y position
- `Volume` (01float): Volume level (0.0 to 1.0)
- `Loop` (bool): If true, loops the video

Restrictions: view, object, global, pocket

PlayVideoWithin

```
PlayVideoWithin(VidFile: vidfile, Start: number, WithinObject:
objnum, Volume: 01float, Loop: bool)
```

Description: Plays a video within the bounds of a specified object. The video is scaled to fit the object’s dimensions.

Parameters:

- `VidFile` (vidfile): Path to the video file
- `Start` (number): Start time in seconds
- `WithinObject` (objnum): ID of the object to play within
- `Volume` (01float): Volume level (0.0 to 1.0)
- `Loop` (bool): If true, loops the video

Restrictions: view, object, global, pocket

StopVideo

```
StopVideo(VideoFile: vidfile)
```

Description: Stops playback of the specified video file.

Parameters:

- `VideoFile` (vidfile): Path to the video file to stop

Restrictions: view, object, global, pocket

StopAllVideos

```
StopAllVideos()
```

Description: Stops all currently playing videos.

Parameters: None

Restrictions: view, object, global, pocket

ShowImage

```
ShowImage(ImageFile: picfile, Left: number, Top: number, Duration:
float, Clickthrough: bool)
```

Description: Displays an image at the specified position for a given duration.

Parameters:

- `ImageFile` (picfile): Path to the image file
- `Left` (number): X position
- `Top` (number): Y position
- `Duration` (float): How long to display (seconds), 0 for permanent
- `Clickthrough` (bool): If true, clicks pass through the image

Restrictions: view, object

ShowImageWithin

```
ShowImageWithin(ImageFile: picfile, Left: number, Top: number,
Duration: float, Clickthrough: bool, WithinObject: number,
HideTarget: bool, Stretch: bool)
```

Description: Displays an image within the bounds of a specified object.

Parameters:

- `ImageFile` (picfile): Path to the image file
- `Left` (number): X offset within the object
- `Top` (number): Y offset within the object
- `Duration` (float): How long to display (seconds)
- `Clickthrough` (bool): If true, clicks pass through
- `WithinObject` (number): ID of the object to display within
- `HideTarget` (bool): If true, hides the target object
- `Stretch` (bool): If true, stretches image to fit object bounds

Restrictions: view, object

ShowObject

```
ShowObject(Object: objnum)
```

Description: Makes a hidden object visible.

Parameters:

- `Object` (objnum): ID of the object to show

Restrictions: view, object, global, pocket

HideObject

```
HideObject(Object: objnum)
```

Description: Hides a visible object.

Parameters:

- `Object` (objnum): ID of the object to hide

Restrictions: view, object, global, pocket

AllowTake

```
AllowTake(Object: objnum)
```

Description: Allows an object to be picked up and placed in a pocket.

Parameters:

- `Object` (objnum): ID of the object

Restrictions: view, object, global, pocket

DisallowTake

```
DisallowTake(Object: objnum)
```

Description: Prevents an object from being picked up.

Parameters:

- `Object` (objnum): ID of the object

Restrictions: view, object, global, pocket

SetVariable

```
SetVariable(Variable: varname, Value: value)
```

Description: Creates or updates a variable with the specified value.

Parameters:

- `Variable` (varname): Name of the variable
- `Value` (value): Value to assign

Restrictions: view, object, global, pocket

DelVariable

```
DelVariable(Variable: varname)
```

Description: Deletes the specified variable.

Parameters:

- `Variable` (varname): Name of the variable to delete

Restrictions: view, object, global, pocket

ClearKeyBuffer

```
ClearKeyBuffer()
```

Description: Clears all accumulated keystrokes from the key buffer.

Parameters: None

Restrictions: view, object, global, pocket

TextBox

```
TextBox(Message: value, Left: number, Top: number, Duration:
float, FontColor: fgcolor, BackColor: bgcolor, FontSize: fontsize,
Bold: bool)
```

Description: Displays a text box with the specified message and styling.

Parameters:

- `Message` (value): Text to display
- `Left` (number): X position
- `Top` (number): Y position
- `Duration` (float): How long to display (seconds)
- `FontColor` (fgcolor): Text color
- `BackColor` (bgcolor): Background color
- `FontSize` (fontsize): Font size
- `Bold` (bool): If true, uses bold text

Restrictions: view, object, global, pocket

TextDialog

```
TextDialog(Message: value)
```

Description: Displays a modal dialog box with the specified message.

Parameters:

- `Message` (value): Text to display

Restrictions: view, object, global, pocket

InputDialog

```
InputDialog(Prompt: value, Variable: varname)
```

Description: Displays a dialog prompting the user for input. The entered value is stored in the specified variable.

Parameters:

- `Prompt` (value): Text prompt to display
- `Variable` (varname): Variable to store the input

Restrictions: view, object, global, pocket

SayText

```
SayText(Message: value)
```

Description: Uses text-to-speech to speak the specified message.

Parameters:

- `Message` (value): Text to speak

Restrictions: view, object, global, pocket

ShowURL

```
ShowURL(URL: value)
```

Description: Opens the specified URL in the default web browser.

Parameters:

- `URL` (value): URL to open

Restrictions: view, object, global, pocket

RunProgram

```
RunProgram(Application: exefile, Parameters: value)
```

Description: Launches an external application with the specified parameters.

Parameters:

- `Application` (exefile): Path to the executable
- `Parameters` (value): Command-line parameters

Restrictions: view, object, global, pocket

ChangeCursor

```
ChangeCursor(Cursor: cursor)
```

Description: Changes the mouse cursor to the specified style.

Parameters:

- `Cursor` (cursor): Cursor style identifier

Restrictions: (no restrictions)

HideMouse

```
HideMouse()
```

Description: Hides the mouse cursor.

Parameters: None

Restrictions: view, object, global, pocket

ShowMouse

```
ShowMouse()
```

Description: Shows the mouse cursor if hidden.

Parameters: None

Restrictions: view, object, global, pocket

HidePockets

```
HidePockets()
```

Description: Hides all pocket UI elements.

Parameters: None

Restrictions: view, object, global, pocket

ShowPockets

```
ShowPockets()
```

Description: Shows all pocket UI elements.

Parameters: None

Restrictions: view, object, global, pocket

Quit

```
Quit()
```

Description: Terminates the GEMS environment.

Parameters: None

Restrictions: view, object, global, pocket

Parameter Types

Type	Description
`viewnum`	View ID selector
`objnum`	Object ID selector
`value`	Text string
`varname`	Variable name
`number`	Integer value
`float`	Decimal number
`01float`	Float between 0.0 and 1.0
`bool`	Boolean (True/False)

Type	Description
`sndfile`	Audio file path
`vidfile`	Video file path
`picfile`	Image file path
`exefile`	Executable file path
`fgcolor`	Foreground color
`bgcolor`	Background color
`fontsize`	Font size
`cursor`	Cursor style
`key`	Keyboard key