Command Module notes

From the The Orx Bounce Demo, or any of your own games, you can bring up the Orx Console with tilde (~).

`Command.ListCommands` lists all commands `Command.ListCommands <prefix>` filters list by prefix.

e.g. `Command.ListCommands Config` only shows commands for the config module

`Command.Help <command>` shows how to use a command.

- Executing a command directly from code:

```
orxCOMMAND_VAR stResult;
orxCommand_Evaluate("Object.Create RedCar", &stResult);
```

`stResult.u64Value` will contain the GUID of the created object.

Of course the string can be loaded from config for something more data-driven.

Core Commands

These commands are registered by the Orx system automatically. See below for the macro orxCOMMAND_REGISTER_CORE_COMMAND that registers these commands.

To register your own commands, write a similar macro or register your command with orxCommand_Register.

```
Camera.Create
Camera.Delete
Camera.Get
Camera.GetID
Camera.GetName
Camera.GetParent
Camera.GetPosition
Camera.GetRotation
Camera.GetZoom
Camera.SetFrustum
Camera.SetParent
Camera.SetPosition
Camera.SetRotation
Camera.SetZoom
Clock.SetFrequency
Clock.SetModifier
Command.Add
Command.AddAlias
```

```
Command.And
Command.AreEqual
Command.Clamp
Command.Compare
Command.CRC
Command.Divide
Command.Evaluate
Command.EvaluateIf
Command.Exit
Command.GetClipboard
Command.GetStringFromID
Command.GetStringID
Command.Help
Command.If
Command.IsGreater
Command.IsLesser
Command.ListAliases
Command.ListCommands
Command.LogAllStructures
Command.Maximum
Command.Minimum
Command.Multiply
Command.Normalize
Command.Not
Command.Or
Command.RemoveAlias
Command.Repeat
Command.Return
Command.SetClipboard
Command.Subtract
Command.Version
Command.XOr
Config.AppendValue
Config.ClearSection
Config.ClearValue
Config.CreateSection
Config.GetCurrentSection
Config.GetListCount
Config.GetOrigin
Config.GetParent
Config.GetRawValue
Config.GetValue
Config.HasSection
Config.HasValue
Config.Load
Config.Reload
Config.Save
Config.SetParent
```

```
Config.SetValue
Console.Echo
Console.Enable
Console.Log
Console.SetColor
Input.EnableSet
Input.GetCurrentSet
Input.GetValue
Input.HasNewStatus
Input.IsActive
Input.IsSetEnabled
Input.ResetValue
Input.SelectSet
Input.SetValue
Locale.GetCurrentLanguage
Locale.GetString
Locale.SelectLanguage
Locale.SetString
Mouse.GetPosition
Mouse.SetCursor
Mouse.SetPosition
Mouse.ShowCursor
Object.AddFX
Object.AddShader
Object.AddSound
Object.AddTrack
Object.Attach
Object.Create
Object.Delete
Object.Detach
Object.Enable
Object.FindNext
Object.GetAlpha
Object.GetAngularVelocity
Object.GetChild
Object.GetClock
Object.GetColor
Object.GetCount
Object.GetCustomGravity
Object.GetGroup
Object.GetHSL
Object.GetHSV
Object.GetID
Object.GetLifeTime
Object.GetName
Object.GetOrigin
Object.GetOwnedChild
Object.GetOwnedSibling
Object.GetOwner
Object.GetParent
Object.GetPivot
```

```
Object.GetPosition
Object.GetRepeat
Object.GetRGB
Object.GetRotation
Object.GetScale
Object.GetSibling
Object.GetSize
Object.GetSpeed
Object.GetText
Object.IsEnabled
Object.IsPaused
Object.Pause
Object.Play
Object.RemoveFX
Object.RemoveShader
Object.RemoveSound
Object.RemoveTrack
Object.SetAlpha
Object.SetAngularVelocity
Object.SetAnim
Object.SetAnimFrequency
Object.SetClock
Object.SetColor
Object.SetCustomGravity
Object.SetGroup
Object.SetHSL
Object.SetHSV
Object.SetLifeTime
Object.SetOrigin
Object.SetOwner
Object.SetParent
Object.SetPitch
Object.SetPivot
Object.SetPosition
Object.SetRepeat
Object.SetRGB
Object.SetRotation
Object.SetScale
Object.SetSize
Object.SetSpeed
Object.SetText
Object.SetVolume
Object.Stop
Render.GetScreenPosition
Render.GetWorldPosition
Resource.AddStorage
Resource.GetPath
Resource.GetTotalPendingOpCount
```

```
Resource.Locate
Resource.ReloadStorage
Resource.RemoveStorage
Screenshot.Capture
Sound.GetBusChild
Sound.GetBusParent
Sound.GetBusPitch
Sound.GetBusSibling
Sound.GetBusVolume
Sound.SetBusParent
Sound.SetBusPitch
Sound.SetBusVolume
Texture.Create
Texture.Delete
Texture.Find
Texture.GetLoadCount
Texture.GetName
Texture.GetSize
Texture.Save
Viewport.AddShader
Viewport.Create
Viewport.Delete
Viewport.Enable
Viewport.EnableShader
Viewport.Get
Viewport.GetCamera
Viewport.GetID
Viewport.GetName
Viewport.GetPosition
Viewport.GetRelativeSize
Viewport.GetSize
Viewport.IsEnabled
Viewport.IsShaderEnabled
Viewport.RemoveShader
Viewport.SetBlendMode
Viewport.SetCamera
Viewport.SetPosition
Viewport.SetRelativePosition
Viewport.SetRelativeSize
Viewport.SetSize
```

Registering Commands

```
/** Registers a command
* @param[in]   _zCommand      Command name
* @param[in]   _pfnFunction   Associated function
* @param[in]   _u32RequiredParamNumber Number of required parameters of the
command
```

```
* @param[in]   _u32OptionalParamNumber Number of optional parameters of the
command
* @param[in]   _astParamList  List of parameters of the command
* @param[in]   _pstResult     Result
* @return      orxSTATUS_SUCCESS / orxSTATUS_FAILURE
*/
extern orxDLLAPI orxSTATUS orxFASTCALL orxCommand_Register(const orxSTRING
_zCommand,
                                                           const
orxCOMMAND_FUNCTION _pfnFunction,
                                                           orxU32
_u32RequiredParamNumber,
                                                           orxU32
_u32OptionalParamNumber,
                                                           const
orxCOMMAND_VAR_DEF *_astParamList,
                                                           const
orxCOMMAND_VAR_DEF *_pstResult);
```

Timeline

Timelines allow for executing commands from the command module in config.

You can ask to push the result of a function as many time as you want but make sure to have the same numbers of push and pop, as usual for a stack.

Here are the current special characters:

```
" is used as a block delimiter, exactly in the same way as in config
> is used for pushing the result of a command and has to be typed before the
command
< is used for popping an element from the stack
^ is used to replace an argument by the GUID of the timeline's owner object
```

Now, if you want to use commands + timelines, let's say for creating a Timer object that will play the beep sounds before a race starts.

In config:

```
; Object template
[Timer]
TrackList = BeepTrack

; TimeLineTrack template
[BeepTrack]
```

```
0 = Object.AddSound ^ RegularBeep
1 = Object.AddSound ^ RegularBeep
2 = Object.AddSound ^ HighPitchBeep
3 = Object.Delete ^
```

In code, create the “Timer” object and the timeline is added to it immediately on creation.

Currently, this is the timeline track template (from CreationSettings.ini):

```
[TimeLineTrackTemplate]
[Float] = "Your timeline event text here" | "Your command"; NB: Float is a time in second (>= 0) after which this timeline event is going to be triggered; If the event is a valid command it'll get executed by the command module;
Loop = true|false;
KeepInCache = true|false; NB: If true, the timeline track will always stay in cache, even if no track of this type is currently in use. Can save time but costs memory. Defaults to false;
```

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/command/commandnotes?rev=1666993273

Last update: **2025/09/30 17:26 (14 months ago)**