



# AstroSynthesis AlienAPI Documentation

© 2019 NBOS Software

# Table of Contents

<b>Part I Change Log</b>	<b>1</b>
<b>Part II AlienAPI Basics</b>	<b>1</b>
<b>1 Command Tags</b>	<b>1</b>
Telling AstroSynthesis about the plug-in	1
#author	2
#bodygen	2
#button	2
#desc	3
#extdesc	3
#fileext	3
#filehandler	3
#generator	4
#menuhotkey	4
#menuinsert	4
#plugin	4
#popmenu	5
<b>2 DynaLoading</b>	<b>5</b>
<b>3 Base Data Types</b>	<b>6</b>
Integer	6
Float	6
Boolean	6
Word	6
String	7
Structures	7
Point	7
Properties	7
X	7
Y	7
Z	7
<b>Part III Reference</b>	<b>7</b>
<b>1 Function Declaration Conventions</b>	<b>7</b>
<b>2 Sector Class</b>	<b>7</b>
<b>Related Functions</b>	<b>8</b>
CreateSector	8
GetCurrentSector	8
<b>Properties</b>	<b>8</b>
Name	8
FileName	8
Notes	8
GMNotes	8
RandomSeed	8
Modified	8
UnitStarMeasureName	9

UnitStarMeasureAbbrev.....	9
UnitStarMeasureRatio.....	9
SizeX .....	9
SizeY .....	9
SizeZ .....	9
Frequency.....	9
Spherical .....	9
VisibleCount.....	9
GridStyle .....	10
GridX .....	10
GridY .....	10
GridZ .....	10
GridTick .....	10
GridTickLabels.....	10
GridColor.....	10
GridsEnabled.....	10
BackgroundColor.....	10
BaseFont.....	11
FilterStyle.....	11
FilterHabitable.....	11
FilterPopulation.....	11
FilterHideUnassignedSubsector.....	11
RenderCameraPos .....	11
RenderCameraTarget.....	11
RenderMessageBig.....	11
RenderMessageSmall.....	11
<b>Methods .....</b>	<b>12</b>
AddCurrentViewAsView.....	12
AddPolitical.....	12
AddResourceFile .....	12
AddRoute.....	12
AddRouteType.....	12
AddSubsector .....	12
AddSystem.....	12
AdjustCameraLatitude .....	12
AdjustDistanceToTarget .....	13
AdjustElevation.....	13
AdjustSelectedDistanceToCamera .....	13
BodyAtXY.....	13
BuildPoliticalList.....	13
BuildRouteTypeList.....	13
CameraOrigin.....	13
CenterCameraOnSelected.....	13
CenterOnOrigin.....	13
ChangeBodyTypeSelected .....	14
ChangeSubsectorName.....	14
CheckFWERefresh.....	14
Clear .....	14
ClearDeleted.....	14
ClearFields.....	14
ComboSlideCamera.....	14
ComboSlideReset.....	14
ComboSlideToSelected.....	14
CreateProximityRoutes.....	15

CreateRouteSegment.....	15
CreateRouteSelected.....	15
CreateSubmapFromSelected.....	15
DeleteRootBody.....	15
DeleteRoute.....	15
DeleteRouteType.....	16
DeleteSelectedRoutes.....	16
DeleteSelectedSystems.....	16
DeleteSubsector.....	16
DeleteSystemFromRoutes.....	16
DeleteView.....	16
DistanceInUnits.....	16
DistanceString.....	16
DynaLoad.....	16
DynaLoadAll.....	17
DynaLoadResource.....	17
DynaUnload.....	17
Export2D.....	17
FindPolitical.....	17
FindResource.....	17
FindRootBodyById.....	17
FindRootBodyByName.....	17
FindRouteType.....	17
FindView.....	18
GenerateSector.....	18
GenerateSubsectors.....	18
GenerateTempFile.....	18
GetField.....	18
GetFieldByIndex.....	18
GetFieldCount.....	18
GetFieldIndex.....	18
GetFieldNameByIndex.....	18
GetResourceName.....	19
GetRoute.....	19
GetSectorBodyCount.....	19
GetSectorPopulation.....	19
GetSelected.....	19
GetSubsector.....	19
GetSubsectorIndexByName.....	19
GetSystem.....	19
GetSystemDataConfigHolder.....	19
ImportSector.....	20
LoadFromFile.....	20
LoadFromTemplate.....	20
LoadView.....	20
MoveSelected.....	20
New.....	20
NewSystem.....	20
NewSystemFromText.....	20
Origin.....	20
PanCameraToBody.....	20
PanCameraToOrigin.....	21
PanCameraToSelected.....	21
Print.....	21

ReassignChild.....	21
RefreshFWESurfaceMap.....	21
ResetCamera.....	21
ResourceCount.....	21
RestoreDefaultOrigins.....	21
RotateCameraAroundTarget.....	21
RotateSelectedAroundCamera.....	22
RouteCount.....	22
RoutesSelected.....	22
SaveGenerateTempFile.....	22
SaveResourceByIndex.....	22
SaveResourceToFile.....	22
SaveToFile.....	22
Search .....	22
SelectAll .....	22
SelectAllHabitable.....	23
SelectBodies.....	23
SelectBody.....	23
SelectedCount.....	23
SetBlazonImage.....	23
SetBlazonSelected.....	23
SetCameraDistance.....	23
SetDefaultSystemDataConfig.....	23
SetDisplayStyleSelected.....	23
SetField .....	23
SetFilters.....	24
SetLabelDistanceSelected.....	24
SetOriginsFromCurrentView.....	24
SetPoliticalSelected.....	24
SetPreviewImage.....	24
SetSphereStyleSelected.....	24
ShowAll .....	24
SlideCameraToSelected.....	24
SubsectorCount.....	25
SystemCount.....	25
TourGuide.....	25
UpdatePoliticalFrom.....	25
UpdateRootAttribs.....	25
UpdateRouteBodies.....	25
UpdateRoutes.....	25
UpdateRoutesForBody.....	25
UpdateRoutesSelected.....	25
UpdateUnits.....	26
UpdateVisibleCount.....	26
<b>3 Body Class .....</b>	<b>26</b>
<b>Related Functions .....</b>	<b>26</b>
CreateBody.....	26
NameFromBodyType.....	26
BodyTypeFromText.....	26
HabitNameFromHabitCode.....	27
<b>Body Type Constants .....</b>	<b>27</b>
BODY_TYPE_ANY.....	27
BODY_TYPE_ASTEROIDBELT.....	27
BODY_TYPE_BLACKHOLE.....	27

BODY_TYPE_BLUEGIANT.....	27
BODY_TYPE_BROWNDWARF.....	27
BODY_TYPE_CLOSEMULT.....	28
BODY_TYPE_DSO.....	28
BODY_TYPE_FLEET.....	28
BODY_TYPE_FREEOBJECT.....	28
BODY_TYPE_GASGIANT.....	28
BODY_TYPE_MARKER.....	28
BODY_TYPE_MEGASTRUCTURE.....	28
BODY_TYPE_MULT.....	28
BODY_TYPE_NEBULA.....	28
BODY_TYPE_NEUTRON.....	28
BODY_TYPE_NOTASSIGNED.....	28
BODY_TYPE_PLANETOID.....	28
BODY_TYPE_PLANETSYSTEM.....	29
BODY_TYPE_REDGIANT.....	29
BODY_TYPE_RING.....	29
BODY_TYPE_SHIP.....	29
BODY_TYPE_SMALLBODY.....	29
BODY_TYPE_STAR.....	29
BODY_TYPE_STATION.....	29
BODY_TYPE_TERRESTRIAL.....	29
BODY_TYPE_WHITEDWARF.....	29
<b>Properties</b> .....	<b>29</b>
Albedo .....	29
AngleAscendingNode.....	30
AnglePeriapsis.....	30
Atmosphere.....	30
AtmosphereNotes.....	30
AxialTilt .....	30
BlazonDisplay .....	30
BlazonImageFile.....	30
BlazonWrap.....	30
BodyCount.....	31
BodyType.....	31
ChildHabitability.....	31
ChildPopulation.....	31
Color .....	31
Density .....	31
Distance .....	31
DSOIntensity.....	31
Eccentricity.....	31
FontStyle.....	32
GMNotes.....	32
Habitability.....	32
Inclination.....	32
LabelColor.....	32
LabelDistance .....	32
Luminosity.....	33
Mass .....	33
Name .....	33
Notes .....	33
Parent .....	33
PoliticalAffiliation.....	33

Population.....	33
PreviewImageFile.....	33
Radius .....	33
RandomSeed.....	34
RetrogradeOrbit.....	34
Rotation .....	34
RouteDistance.....	34
Selected .....	34
Spectral .....	34
SphereInfluence.....	34
SphereInfluenceColor.....	35
SphereInfluenceSize.....	35
Temp .....	35
TimeOffset.....	35
TypeID .....	35
Visible .....	35
Water .....	35
X .....	35
Y .....	36
Z .....	36
<b>Methods .....</b>	<b>36</b>
AddChild .....	36
ChildrenCount.....	36
ClearChildrenNames.....	36
ClearFields.....	36
CopyTo .....	36
DefaultTextureFile.....	36
DeleteChildren .....	37
DistanceFromParentStar.....	37
FindChildIndex.....	37
FormatChildPopulation.....	37
FormatPopulation.....	37
FormatSystemPopulation.....	37
FormatYearLength.....	37
FWEFileName.....	37
FWEUpdateTerrain.....	38
GetAtmElement.....	38
GetAtmElementCount.....	38
GetAtmElementPercent.....	38
GetBodyCount.....	38
GetChild .....	38
GetChildPopulation.....	38
GetField .....	38
GetFieldByIndex.....	38
GetFieldCount.....	39
GetFieldIndex.....	39
GetFieldNameByIndex.....	39
GetFWECOLORGroup.....	39
GetGeostationary.....	39
GetHillSphere.....	39
GetMaxChildHabitability.....	39
GetOrbit .....	39
GetOrbitalParent.....	40
GetParentBody.....	40

GetParentStar.....	40
GetRootBody.....	40
GetSystemPopulation.....	40
GreenhouseEffectK.....	40
HasParent.....	40
InsertBefore.....	40
ModifySystem.....	41
ResetAtmElements.....	41
SetAtmElement.....	41
SetChildDesignations.....	41
SetField.....	41
SetFWColorGroup.....	41
SetStarMetrics.....	41
SetSystemDataConfig.....	41
SortChildrenByDistance.....	42
SystemModified.....	42
TempAt.....	42
TextureExists.....	42
TextureName.....	42
TextureNameFromSpectral.....	42
TexturePath.....	42
UpdateRootBody.....	42
YearLength.....	43
<b>4 Route Class .....</b>	<b>43</b>
<b>Related Functions .....</b>	<b>43</b>
CreateRoute.....	43
<b>Properties .....</b>	<b>43</b>
Name.....	43
RouteType.....	43
Red.....	43
Green.....	43
Blue.....	43
LineWidth.....	43
LineStyle.....	44
Stipple.....	44
<b>Methods .....</b>	<b>44</b>
CopyTo.....	44
AddSystem.....	44
AddWaypoint.....	44
SelectObject.....	44
ClickedOn.....	44
WaypointCount.....	45
GetWaypoint.....	45
DeleteWaypoint.....	45
OnRoute.....	45
SetField.....	45
GetField.....	45
GetFieldIndex.....	45
GetFieldByIndex.....	45
GetFieldNameByIndex.....	45
GetFieldCount.....	45
ClearFields.....	46
<b>5 RouteWaypoint Class .....</b>	<b>46</b>



<b>Related Functions</b> .....	<b>46</b>
CreateWaypoint.....	46
<b>Properties</b> .....	<b>46</b>
Body .....	46
id .....	46
Pos .....	46
X .....	46
Y .....	46
Z .....	46
<b>Methods</b> .....	<b>47</b>
<b>6 Subsector</b> .....	<b>47</b>
<b>Related Functions</b> .....	<b>47</b>
CreateSubsector.....	47
<b>Properties</b> .....	<b>47</b>
Name .....	47
Color .....	47
ShowLabel .....	47
Shape .....	47
Visible .....	47
Center .....	48
X .....	48
Y .....	48
Z .....	48
SizeX .....	48
SizeY .....	48
SizeZ .....	48
RotateX .....	48
RotateY .....	48
RotateZ .....	49
ShowGrid.....	49
GridX .....	49
GridY .....	49
GridZ .....	49
GridTick .....	49
TickLabels.....	49
<b>Methods</b> .....	<b>49</b>
CopyTo .....	49
inSubsector.....	50
SetField .....	50
GetField .....	50
GetFieldIndex.....	50
GetFieldNameByIndex.....	50
GetFieldCount.....	50
ClearFields.....	50
GetFieldByIndex.....	50
<b>7 SystemDataConfig Class</b> .....	<b>50</b>
<b>Related Functions</b> .....	<b>51</b>
CreateSystemDataConfig.....	51
Body:SetSystemDataConfig.....	51
<b>Properties</b> .....	<b>51</b>
<b>Methods</b> .....	<b>51</b>
AddField .....	51
Clear .....	52

CopyFrom.....	52
Count .....	52
DeleteField.....	52
FindField.....	52
MoveUpDown.....	52
<b>8 SystemDataConfigHolder .....</b>	<b>52</b>
<b>Methods .....</b>	<b>52</b>
SetConfigForType.....	53
<b>9 Orbit Class .....</b>	<b>53</b>
<b>Related Functions .....</b>	<b>53</b>
<b>Properties .....</b>	<b>53</b>
CurrentTime.....	53
PositionX.....	53
PositionY.....	53
PositionZ.....	53
<b>Methods .....</b>	<b>53</b>
SetTime .....	53
<b>10 Movie Functions .....</b>	<b>54</b>
<b>StartMovie .....</b>	<b>54</b>
<b>AddMovieFrame .....</b>	<b>54</b>
<b>FinishMovie .....</b>	<b>54</b>
<b>11 InspirationPad Class .....</b>	<b>54</b>
<b>Related Functions .....</b>	<b>54</b>
CreateIPadTable.....	54
<b>Properties .....</b>	<b>54</b>
LastResultCount.....	54
<b>Methods .....</b>	<b>54</b>
AddVariableDef.....	55
AddDefinition.....	55
ClearAllVars.....	55
ShuffleAll.....	55
Go .....	55
GetResult.....	55
<b>12 Graphical User Interface Classes .....</b>	<b>56</b>
<b>Dialog Window .....</b>	<b>56</b>
Related Functions.....	56
NewDialogWindow.....	56
Properties.....	56
Top .....	56
Left .....	56
Width .....	56
Height .....	56
Centered .....	56
Caption .....	56
Methods .....	56
AddTextEdit .....	57
AddLabel .....	57
AddCheckbox .....	57
AddRadioGroup.....	57
AddMemo .....	57
AddListbox .....	57
AddDropList .....	57

AddCombo .....	57
AddGroupBox .....	57
AddTrackBar .....	58
AddImage .....	58
AddFileOpen .....	58
AddFileSave .....	58
AddColorSelector .....	58
ShowModal .....	58
<b>Widget .....</b>	<b>58</b>
Properties .....	58
Top .....	59
Left .....	59
Width .....	59
Height .....	59
Text .....	59
Caption .....	59
Checked .....	59
Columns .....	59
Color .....	59
FileMask .....	59
FileName .....	60
MinValue .....	60
MaxValue .....	60
Position .....	60
ImageFileName .....	60
Fit .....	60
Methods .....	60
AddItem .....	60
SetPosition .....	61
<b>13 GenerateOptions Class .....</b>	<b>61</b>
<b>Properties .....</b>	<b>61</b>
BodyType .....	61
GenContents .....	61
GenPopulations .....	61
NameGenerator .....	61
<b>Methods .....</b>	<b>61</b>
<b>14 FWIColorGroup .....</b>	<b>62</b>
<b>Properties .....</b>	<b>62</b>
AtmColor .....	62
Beach .....	62
DeepWater .....	62
Dry .....	62
GenAtm .....	62
HighLand .....	62
Ice .....	62
LowLand .....	63
LowWater .....	63
<b>15 Utility Functions .....</b>	<b>63</b>
<b>AppVersionBuild .....</b>	<b>63</b>
<b>AppVersionMajor .....</b>	<b>63</b>
<b>AppVersionMinor .....</b>	<b>63</b>
<b>AstroDirectory .....</b>	<b>63</b>
<b>ClearMap .....</b>	<b>63</b>

DeselectEditingBody .....	64
EditBody .....	64
EditingBody .....	64
FileHandlerFileName .....	64
FreeObject .....	64
GetGenerateOptions .....	64
GetKey .....	64
isNull .....	65
Pause .....	65
PluginDirectory .....	65
ProcessMessages .....	65
RefreshScene .....	65
ShowSystemDiagram .....	65
UpdateCaption .....	65
<b>16 AstroSearcher Class .....</b>	<b>66</b>
<b>Related Functions .....</b>	<b>66</b>
CreateAstroSearcher.....	66
<b>Properties .....</b>	<b>66</b>
ErrorMsg.....	66
<b>Methods .....</b>	<b>67</b>
Search .....	67
GetResultCount.....	67
GetResult.....	67
 <b>Index</b>	 <b>0</b>

# 1 Change Log

11/23/12

Added CreateSystem() function.

3/10/12

Added AstroSearcher class info for Astro 3.0

7/25/06

Added data type info, SystemDataConfigHolder, FWIColorGroup

5/17/06

Added Utility Functions listings and GenerateOptions class.

5/8/06

Added SystemDataConfig object reference

4/17/06

Added IPad, DialogWindow, Widget declarations

4/15/06

Added Route, RouteWaypoint, Subsector, Searcher, Movie, Orbit declarations.

4/12/06

Added Sector Class methods and properties

## 2 AlienAPI Basics

### 2.1 Command Tags

#### 2.1.1 Telling AstroSynthesis about the plug-in

When AstroSynthesis loads, it looks at the various command tags set within each plug-in. These tags tell the program what type plug-in the script represents and various bits of data about the plug-in.

Command tags are simply formatted lines put into the script. They are preceded with '#' symbols. Here's some examples:

```
#plugin      Quick Generate
#author      NBOS
#desc        Generates random systems until the Escape key is pressed
#button      ExampleStuff\QuickGenerate.bmp
```

There are several different types of plug-ins. Scripts need to have command tags telling the program that the script is one of the available types, or the plug-in won't be loaded. The types are:

- Program Feature. These plug-ins add menu items to the program, essentially creating a new feature in the program. Feature plug-ins are defined with the [#plugin](#)<sup>[4]</sup> tag.
- Pop-up Menu Feature. These plug-ins add an item to the pop-up menu that is displayed when a body is right-clicked. These are defined with the [#popmenu](#)<sup>[5]</sup> tag.
- Generator plug-in. These are System Generator plug-ins. They work like normal Program Feature plug-ins, but are automatically placed on the File - New menu. These are defined with the [#generator](#)<sup>[4]</sup> tag.
- Body Generator. These are system generator plug-ins that are made available on the New System window. These are used to generate single systems, not entire sectors. Defined with the [#bodygen](#)<sup>[2]</sup> tag.
- File Handler. These plug-ins allow script creators to tell AstroSynthesis how to open files other than the normal '.sector' file. These plug-ins are defined with the [#filehandler](#)<sup>[3]</sup> tag.

### 2.1.2 #author

```
#author name
```

The #author tag lets you tell the program who created the script. *name* is your name, and is displayed in the Plug-in manager when the plug-in is selected.

### 2.1.3 #bodygen

```
#bodygen name
```

This tells the program that the plug-in is a single body generator. *name* is the name of the plug-in. Body generators are placed on the list of generators available on the New System window.

### 2.1.4 #button

```
#button file
```

The #button tag lets you define a toolbar button image to use for this plug-in. Useful for Program Feature plug-ins. *file* is a 16x16 or 32x16 256 color bitmap file. The path should be relative to the program's 'Plugins' directory. If 16x16 in size, that image is placed on the toolbar as a button that can be used to call the plug-in. If 32x16, the first 16x16 block is the toolbar button, and the second 16x16 block is a grayed out bitmap to display while plug-ins are running.

Example:

```
#button ExampleStuff\QuickGenerate.bmp
```

### 2.1.5 #desc

```
#desc description
```

This lets you define a short description for the plug-in.

Example:

```
#desc      Generates random systems until the Escape key is pressed
```

### 2.1.6 #extdesc

```
#extdesc description
```

For file handler plug-ins, this is a short descriptive name of the file format. This is what is displayed in the file types drop down box in the File Open dialog window when you open a file.

Example:

```
#filehandler      Example File Handler
#fileext          .png
#extdesc          PNG Image Files
#desc             Example of how to register a file extension
```

### 2.1.7 #fileext

```
#fileext extension
```

For file handler plug-ins, this lets you define a file extension to handle. extension is the file extension with preceding period, such as ".xml" or ".dat".

Example:

```
#filehandler      Example File Handler
#fileext          .png
#extdesc          PNG Image Files
#desc             Example of how to register a file extension
```

### 2.1.8 #filehandler

```
#filehandler name
```

This tells the program this is a file handler plug-in. Name is the name of the plug-in. File handler plug-ins are called when someone tries to open a file of a specified type.

Example:

```
#filehandler      Example File Handler
#fileext          .png
#extdesc          PNG Image Files
```

```
#desc Example of how to register a file extension
```

### 2.1.9 #generator

```
#generator name
```

This tells the program this is a Sector generator. Name is the name of the generator. These plug-ins are added to the File - New menu in the program.

### 2.1.10 #menuhotkey

```
#menuhotkey hot-key
```

For Program Feature plug-ins, this lets you define a menu hot-key (or combination) to associate with the menu item created for the plug-in.

Examples:

```
#menuhotkey F5
#menuhotkey Ctrl+B
#menuhotkey Alt+F1
#menuhotkey Alt+W
#menuhotkey Shift+F12
```

### 2.1.11 #menuinsert

```
#menuinsert location
```

For program feature plug-ins, this lets you specify where to put the menu item for this plug-in. By default all plug-ins are placed on the Plugins menu. By setting values for #menuinsert, you can change this. The positions are defined by the menu number, from left to right. That is, the File menu is 1, Sector is 2, etc. If you use a single number, the plug-in is added to the bottom of that menu's list of items. If you use two numbers separated by a space in the #menuinsert tag, the plug-in is placed at the position of the second number within the menu defined by the first number.

Examples

'places the plug-in as the 8th entry of the first menu. In a default installation, this is right after the Import entry in the File menu.

```
#menuinsert 1 8
```

'places the plug-in at the end of the 4th menu, typically the Actions menu.

```
#menuinsert 4
```

### 2.1.12 #plugin

```
#plugin name
```

This tells the program the script is a Program Feature plug-in. These plug-ins create menu items that can be used to call the plug-in. Name is the name of the plug-in, and is what is displayed on the menu.



Example:

```
#plugin XML Import
```

### 2.1.13 #popmenu

```
#popmenu name
```

This tag tells the program the plug-in is a Pop Up Menu plug-in. These plug-ins are made available to run as items in the pop-up menu displayed when you right-click on a body on the map or in the System Data tree. *Name* is the name of the plug-in, and is what is displayed on the menu.

## 2.2 DynaLoading

To allow it to handle very large sector maps, AstroSynthesis only keeps in memory the star system data it needs to be able to display the 3D map. In practice, this means that at any given time, only the root body of a system is guaranteed to be loaded in memory. If you want to access the contents of a system (planets, moons, and other child bodies), you *must* use the *DynaLoad* methods to tell AstroSynthesis to load the system data from the .sector file.

To do this, you can use the *DynaLoad* method of the Sector class. Call this method by passing in the system's root body into the *DynaLoad* method.

```
sector = GetCurrentSector()  
b = sector.GetBody( 0) 'get the first body in the sector  
sector.DynaLoad( b)
```

You can also first test to see if a body is loaded or not before calling *DynaLoad* by checking the body's *Loaded* attribute.

```
sector = GetCurrentSector()  
b = sector.GetBody( 0) 'get the first body in the sector  
If Not b.Loaded Then  
    sector.DynaLoad( b)  
End If
```

After DynaLoading a system from the file, you may want to free that memory if you are done looking at that star system. Otherwise, if you cycle through every system on the map and DynaLoad it, you may end up using a tremendous amount of RAM (or just running out of RAM). To release the system from memory, call the Sector's *DynaUnload* method.

```
sector = GetCurrentSector()  
b = sector.GetBody( 0) 'get the first body in the sector  
If Not b.Loaded Then  
    sector.DynaLoad( b)  
End If  
  
'... do the rest of your script
```

```
sector.DynaUnload( b) 'Only do this if you HAVE NOT made any changes to the sys
```

It's important that you **do not** DynaUnload a system if you've made changes to it. That is, if you added new bodies to the system, changed some names or populations, added custom fields, etc. If you Unload a system that has changes, those changes **will be lost**.

A good practice is to only DynaUnload a body that you previously loaded in your script, and thus know for sure whether or not it has been changed. You'd only unload systems you've loaded yourself in your script so that you do not accidentally unload a system that was modified by the user via the program's user interface. To do this, use a script variable to track whether or not the body was loaded by your script and thus can be unloaded. But remember - **don't unload a system that you make any changes to**.

```
sector = GetCurrentSector()
b = sector.GetBody( 0) 'get the first body in the sector

bUnload = False
If Not b.Loaded Then
    sector.DynaLoad b
    bUnload = True 'make a note that this body needs to be unloaded
End If

'... do the rest of your script

'if your script had loaded the body, unload it
If bUnload Then
    sector.DynaUnload b 'Only do this if you HAVE NOT made any changes to the
End If
```

## 2.3 Base Data Types

### 2.3.1 Integer

AlienAPI integers are 32-bit (4 byte) integers.

### 2.3.2 Float

Floats, as used by the AlienAPI, are 8-byte double precision floating point numbers.

### 2.3.3 Boolean

Boolean variables can contain either True or False

### 2.3.4 Word

A Word variable holds a 16-bit (2 byte) integer.

## 2.3.5 String

String variables are used to hold Text. Strings as used by the AlienAPI are not null terminated.

## 2.3.6 Structures

### 2.3.6.1 Point

The Point type is a structure that defines a location in 3D space

#### 2.3.6.1.1 Properties

##### 2.3.6.1.1.1 X

```
X: Float
```

##### 2.3.6.1.1.2 Y

```
Y: Float
```

##### 2.3.6.1.1.3 Z

```
Z: Float
```

## 3 Reference

### 3.1 Function Declaration Conventions

```
Function Name( ParameterName: ParameterType; ParameterName: ParameterType; ...):
```

If nothing is specified for *ReturnType*, then no value is returned.

### 3.2 Sector Class

## 3.2.1 Related Functions

### 3.2.1.1 CreateSector

```
CreateSector(): Sector
```

Creates a new, in-memory Sector map object. The create sector map is not displayed. It is held strictly in memory.

### 3.2.1.2 GetCurrentSector

```
GetCurrentSector(): Sector;
```

## 3.2.2 Properties

### 3.2.2.1 Name

```
Name: String
```

### 3.2.2.2 FileName

```
FileName: String
```

### 3.2.2.3 Notes

```
Notes: String
```

### 3.2.2.4 GMNotes

```
GMNotes: String
```

### 3.2.2.5 RandomSeed

```
RandomSeed: Integer
```

### 3.2.2.6 Modified

```
Modified: Boolean
```

**3.2.2.7 UnitStarMeasureName**

```
UnitStarMeasureName: String
```

**3.2.2.8 UnitStarMeasureAbbrev**

```
UnitStarMeasureAbbrev: String
```

**3.2.2.9 UnitStarMeasureRatio**

```
UnitStarMeasureRatio: Float
```

**3.2.2.10 SizeX**

```
SizeX: Float
```

**3.2.2.11 SizeY**

```
SizeY: Float
```

**3.2.2.12 SizeZ**

```
SizeZ: Float
```

**3.2.2.13 Frequency**

```
Frequency: Float
```

**3.2.2.14 Spherical**

```
Spherical: Boolean
```

**3.2.2.15 VisibleCount**

```
VisibleCount: Integer
```

**3.2.2.16 GridStyle**

```
GridStyle: Integer
```

**3.2.2.17 GridX**

```
GridX: Float
```

**3.2.2.18 GridY**

```
GridY: Float
```

**3.2.2.19 GridZ**

```
GridZ: Float
```

**3.2.2.20 GridTick**

```
GridTick: Float
```

**3.2.2.21 GridTickLabels**

```
GridTickLabels: Boolean
```

**3.2.2.22 GridColor**

```
GridColor: Integer
```

**3.2.2.23 GridsEnabled**

```
GridsEnabled: Boolean
```

**3.2.2.24 BackgroundColor**

```
BackgroundColor: Integer
```

**3.2.2.25 BaseFont**

```
BaseFont: String
```

**3.2.2.26 FilterStyle**

```
FilterStyle: Integer
```

**3.2.2.27 FilterHabitable**

```
FilterHabitable: Integer
```

**3.2.2.28 FilterPopulation**

```
FilterPopulation: Integer
```

**3.2.2.29 FilterHideUnassignedSubsector**

```
FilterHideUnassignedSubsector: Boolean
```

**3.2.2.30 RenderCameraPos**

```
RenderCameraPos: Point
```

**3.2.2.31 RenderCameraTarget**

```
RenderCameraTarget: Point
```

**3.2.2.32 RenderMessageBig**

```
RenderMessageBig: String
```

**3.2.2.33 RenderMessageSmall**

```
RenderMessageSmall: String
```

### 3.2.3 Methods

#### 3.2.3.1 AddCurrentViewAsView

```
AddCurrentViewAsView( sName: String)
```

#### 3.2.3.2 AddPolitical

```
AddPolitical( s: String): Integer
```

#### 3.2.3.3 AddResourceFile

```
AddResourceFile( sFile: String): Boolean
```

#### 3.2.3.4 AddRoute

```
AddRoute( r: Route): Integer
```

#### 3.2.3.5 AddRouteType

```
AddRouteType( s: String): Integer
```

#### 3.2.3.6 AddSubsector

```
AddSubsector( ss: Subsector): Integer
```

#### 3.2.3.7 AddSystem

```
AddSystem( b: Body): Integer
```

#### 3.2.3.8 AdjustCameraLatitude

```
AdjustCameraLatitude( n: Float)
```



### 3.2.3.9 AdjustDistanceToTarget

```
AdjustDistanceToTarget( f: Float)
```

### 3.2.3.10 AdjustElevation

```
AdjustElevation( f: Float)
```

### 3.2.3.11 AdjustSelectedDistanceToCamera

```
AdjustSelectedDistanceToCamera( f: Float)
```

### 3.2.3.12 BodyAtXY

```
BodyAtXY( x: Integer; y: Integer): Body
```

### 3.2.3.13 BuildPoliticalList

```
BuildPoliticalList( bClear: Boolean)
```

### 3.2.3.14 BuildRouteTypeList

```
BuildRouteTypeList( bClear: Boolean)
```

### 3.2.3.15 CameraOrigin

```
CameraOrigin(): Point
```

### 3.2.3.16 CenterCameraOnSelected

```
CenterCameraOnSelected()
```

### 3.2.3.17 CenterOnOrigin

```
CenterOnOrigin()
```

**3.2.3.18 ChangeBodyTypeSelected**

```
ChangeBodyTypeSelected( nType: Integer)
```

**3.2.3.19 ChangeSubsectorName**

```
ChangeSubsectorName( ss: Subsector; sName: String)
```

**3.2.3.20 CheckFWERefresh**

```
CheckFWERefresh( b: Body)
```

**3.2.3.21 Clear**

```
Clear()
```

**3.2.3.22 ClearDeleted**

```
ClearDeleted()
```

**3.2.3.23 ClearFields**

```
ClearFields()
```

**3.2.3.24 ComboSlideCamera**

```
ComboSlideCamera( ptC: Point; ptT: Point; nSteps: Integer; nStep: Integer)
```

**3.2.3.25 ComboSlideReset**

```
ComboSlideReset( nSteps: Integer; nStep: Integer)
```

**3.2.3.26 ComboSlideToSelected**

```
ComboSlideToSelected( nSteps: Integer; nStep: Integer)
```

### 3.2.3.27 CreateProximityRoutes

```
CreateProximityRoutes( nType: Integer; nMin: Float; nMax: Float; nMR: Float; sRoute
```

### 3.2.3.28 CreateRouteSegment

```
CreateRouteSegment( b1: Body; b2: Body; sType: String; sName: String; nColor: Integer
```

### 3.2.3.29 CreateRouteSelected

```
CreateRouteSelected( sType: String; sName: String; Red: Float; Green: Float; Blue: Float
```

### 3.2.3.30 CreateSubmapFromSelected

```
CreateSubmapFromSelected( newsector: Sector; nSize: Float; nShape: Integer)
```

Copies a section of a Sector map into another Sector map, based around the location of the first selected system on the map.

*newsector* is the destination Sector object that will be copied to. All systems within *nSize* light years of the first selected system on a map are copied into *newsector*, and the center of the map is repositioned so that the selected system in the source sector is located at (0,0,0). Since *nSize* specifies the distance from a selected system, the resultant map will be *nSize* x 2 across.

*nShape* represents the selection style when determining which nearby systems to include in the sub-map. If *nShape* is 0, then all systems within a cube whose edges are *nSize* from the selected system are included. If *nShape* is 1, then only those systems within *nSize* in distance from the selected system are included (i.e., a spherical selection of all systems within *nSize* radius).

### 3.2.3.31 DeleteRootBody

```
DeleteRootBody( b: Body)
```

### 3.2.3.32 DeleteRoute

```
DeleteRoute( n: Integer)
```

**3.2.3.33 DeleteRouteType**

```
DeleteRouteType( sRouteType: String)
```

**3.2.3.34 DeleteSelectedRoutes**

```
DeleteSelectedRoutes()
```

**3.2.3.35 DeleteSelectedSystems**

```
DeleteSelectedSystems()
```

**3.2.3.36 DeleteSubsector**

```
DeleteSubsector(n: Integer)
```

**3.2.3.37 DeleteSystemFromRoutes**

```
DeleteSystemFromRoutes( b: Body)
```

**3.2.3.38 DeleteView**

```
DeleteView( sName: String)
```

**3.2.3.39 DistanceInUnits**

```
DistanceInUnits( n: Float): Float
```

**3.2.3.40 DistanceString**

```
DistanceString( n: Float; bShowUnit: Boolean): String
```

**3.2.3.41 DynaLoad**

```
DynaLoad( b: Body)
```

**3.2.3.42 DynaLoadAll**

```
DynaLoadAll()
```

**3.2.3.43 DynaLoadResource**

```
DynaLoadResource( r: ImageResource)
```

**3.2.3.44 DynaUnload**

```
DynaUnload( b: Body)
```

**3.2.3.45 Export2D**

```
Export2D(sFile: String; colorstyle: Integer; bRoutes: Boolean; bScale: Boolean; bGr
```

**3.2.3.46 FindPolitical**

```
FindPolitical( s: String): Integer
```

**3.2.3.47 FindResource**

```
FindResource( sResource: String): Integer
```

**3.2.3.48 FindRootBodyById**

```
FindRootBodyById( sID: String): Body
```

**3.2.3.49 FindRootBodyByName**

```
FindRootBodyByName( s: String): Body
```

**3.2.3.50 FindRouteType**

```
FindRouteType( s: String): Integer
```

**3.2.3.51 FindView**

```
FindView( sName: String): Integer
```

**3.2.3.52 GenerateSector**

```
GenerateSector( nDistribution: Integer; nOptions: Integer; bPopulations: Boolean; S
```

**3.2.3.53 GenerateSubsectors**

```
GenerateSubsectors( nX: Integer; nY: Integer; nZ: Integer; c: Integer; sNameTable:
```

**3.2.3.54 GenerateTempFile**

```
GenerateTempFile(): String
```

**3.2.3.55 GetField**

```
GetField( sName: String): String
```

**3.2.3.56 GetFieldByIndex**

```
GetFieldByIndex( n: Integer): String
```

**3.2.3.57 GetFieldCount**

```
GetFieldCount(): Integer
```

**3.2.3.58 GetFieldIndex**

```
GetFieldIndex( sName: String): Integer
```

**3.2.3.59 GetFieldNameByIndex**

```
GetFieldNameByIndex( n: Integer): String
```

**3.2.3.60 GetResourceName**

```
GetResourceName( n: Integer): String
```

**3.2.3.61 GetRoute**

```
GetRoute(n: Integer): Route
```

**3.2.3.62 GetSectorBodyCount**

```
GetSectorBodyCount(): Float
```

**3.2.3.63 GetSectorPopulation**

```
GetSectorPopulation(): Float
```

**3.2.3.64 GetSelected**

```
GetSelected( n: Integer): Body
```

**3.2.3.65 GetSubsector**

```
GetSubsector(n: Integer): Subsector
```

**3.2.3.66 GetSubsectorIndexByName**

```
GetSubsectorIndexByName( s: String): Integer
```

**3.2.3.67 GetSystem**

```
GetSystem( n: Integer): Body
```

**3.2.3.68 GetSystemDataConfigHolder**

```
GeSystemDataConfigHolder(): SystemDataConfigHolder
```

**3.2.3.69 ImportSector**

```
ImportSector( sFile: String; replace: Integer; placement: Integer; x: Float; y: Flo
```

**3.2.3.70 LoadFromFile**

```
LoadFromFile( sFile: String)
```

**3.2.3.71 LoadFromTemplate**

```
LoadFromTemplate( sFile: String)
```

**3.2.3.72 LoadView**

```
LoadView( sName: String)
```

**3.2.3.73 MoveSelected**

```
MoveSelected( x: Float; y: Float; z: Float)
```

**3.2.3.74 New**

```
New()
```

**3.2.3.75 NewSystem**

```
NewSystem( nType: Integer): Body
```

**3.2.3.76 NewSystemFromText**

```
NewSystemFromText( sType: String; sNameTable: String): Body
```

**3.2.3.77 Origin**

```
Origin(): Point
```

**3.2.3.78 PanCameraToBody**

```
PanCameraToBody( b: Body; nSteps: Integer; nStep: Integer)
```



**3.2.3.79 PanCameraToOrigin**

```
PanCameraToOrigin( nSteps: Integer; nStep: Integer)
```

**3.2.3.80 PanCameraToSelected**

```
PanCameraToSelected( nSteps: Integer; nStep: Integer)
```

**3.2.3.81 Print**

```
Print( bPreview: Boolean; nPagesAcross: Float; colorstyle: Integer; bRoutes: Boolean)
```

**3.2.3.82 ReassignChild**

```
ReassignChild( bSource: Body; bTarget: Body)
```

**3.2.3.83 RefreshFWESurfaceMap**

```
RefreshFWESurfaceMap( b: Body)
```

**3.2.3.84 ResetCamera**

```
ResetCamera()
```

**3.2.3.85 ResourceCount**

```
ResourceCount(): Integer
```

**3.2.3.86 RestoreDefaultOrigins**

```
RestoreDefaultOrigins()
```

**3.2.3.87 RotateCameraAroundTarget**

```
RotateCameraAroundTarget( xdeg: Float; ydeg: Float; zdeg: Float)
```

**3.2.3.88 RotateSelectedAroundCamera**

```
RotateSelectedAroundCamera( dx: Float; dy: Float; dz: Float)
```

**3.2.3.89 RouteCount**

```
RouteCount(): Integer
```

**3.2.3.90 RoutesSelected**

```
RoutesSelected(): Boolean
```

**3.2.3.91 SaveGenerateTempFile**

```
SaveGenerateTempFile( sFile: String)
```

**3.2.3.92 SaveResourceByIndex**

```
SaveResourceByIndex(n: Integer; sFile: String): Boolean
```

**3.2.3.93 SaveResourceToFile**

```
SaveResourceToFile( sResource: String; sFile: String): Boolean
```

**3.2.3.94 SaveToFile**

```
SaveToFile( sFile: String; bOverwrite: Boolean)
```

**3.2.3.95 Search**

```
Search( src: Searcher): Integer
```

**3.2.3.96 SelectAll**

```
SelectAll( bSelected: Boolean)
```

**3.2.3.97 SelectAllHabitable**

```
SelectAllHabitable( n: Integer)
```

**3.2.3.98 SelectBodies**

```
SelectBodies( x: Integer; y: Integer; bAccumulate: Boolean): Integer
```

**3.2.3.99 SelectBody**

```
SelectBody( b: Body)
```

**3.2.3.100 SelectedCount**

```
SelectedCount(): Integer
```

**3.2.3.101 SetBlazonImage**

```
SetBlazonImage( b: Body; sFile: String)
```

**3.2.3.102 SetBlazonSelected**

```
SetBlazonSelected( bUseBlazon: Boolean; bWrap: Boolean; sFile: String)
```

**3.2.3.103 SetCameraDistance**

```
SetCameraDistance( d: Float)
```

**3.2.3.104 SetDefaultSystemDataConfig**

```
SetDefaultSystemDataConfig( sdh: SystemDataConfigHolder)
```

**3.2.3.105 SetDisplayStyleSelected**

```
SetDisplayStyleSelected( StarColor: Integer; LabelColor: Integer; FontBold: Boolean)
```

**3.2.3.106 SetField**

```
SetField( sName: String; sVal: String)
```

**3.2.3.107 SetFilters**

```
SetFilters()
```

**3.2.3.108 SetLabelDistanceSelected**

```
SetLabelDistanceSelected( nLabelDistance: Float; nRouteDistance: Float)
```

**3.2.3.109 SetOriginsFromCurrentView**

```
SetOriginsFromCurrentView()
```

**3.2.3.110 SetPoliticalSelected**

```
SetPoliticalSelected( s: String)
```

**3.2.3.111 SetPreviewImage**

```
SetPreviewImage( b: Body; sFile: String)
```

**3.2.3.112 SetSphereStyleSelected**

```
SetSphereStyleSelected( SphereStyle: Integer; SphereColor: Integer; SphereSize: FL
```

**3.2.3.113 ShowAll**

```
ShowAll()
```

**3.2.3.114 SlideCameraToSelected**

```
SlideCameraToSelected( nSteps: Integer; nStep: Integer)
```

**3.2.3.115 SubsectorCount**

```
SubsectorCount(): Integer
```

**3.2.3.116 SystemCount**

```
SystemCount(): Integer
```

**3.2.3.117 TourGuide**

```
TourGuide()
```

**3.2.3.118 UpdatePoliticalFrom**

```
UpdatePoliticalFrom( b: Body)
```

**3.2.3.119 UpdateRootAttribs**

```
UpdateRootAttribs()
```

**3.2.3.120 UpdateRouteBodies**

```
UpdateRouteBodies()
```

**3.2.3.121 UpdateRoutes**

```
UpdateRoutes()
```

**3.2.3.122 UpdateRoutesForBody**

```
UpdateRoutesForBody( b: Body)
```

**3.2.3.123 UpdateRoutesSelected**

```
UpdateRoutesSelected()
```

### 3.2.3.124 UpdateUnits

```
UpdateUnits()
```

### 3.2.3.125 UpdateVisibleCount

```
UpdateVisibleCount()
```

## 3.3 Body Class

The Body class encapsulates all the properties and methods for each body (star, planet, black hole, etc) on your system. Each body on your map is represented in the system by a Body class object. All body types contain the same properties and methods - though not all properties or methods will apply to all types.

### 3.3.1 Related Functions

#### 3.3.1.1 CreateBody

```
Function CreateBody( sType: String): Body
```

Creates and returns a body object. *sType* is either an empty string, which returns a body of unassigned type, or one of the strings as listed in the `BodyTypeFromText` function.

#### 3.3.1.2 NameFromBodyType

```
Function NameFromBodyType( nBodyType: integer): String
```

Returns a string description of a given integer body type value.

#### 3.3.1.3 BodyTypeFromText

```
Function BodyTypeFromText( sBodyType: String): Integer
```

Returns the integer body type ID value for a given textual description of a body type. *sBodyType* may be one of the following:

```
star
multiple star system
close multiple system
black hole
space station
free object
neutron star
nebula
white dwarf
```

```
red giant
blue giant
brown dwarf
deep space object
terrestrial
gas giant
asteroid belt
planetoid
ring
planet system
small body
mega structure
fleet
ship
marker
```

#### 3.3.1.4 HabitNameFromHabitCode

```
Function HabitNameFromHabitCode( n: Integer): String
```

Returns the text description of a habitability code. This will return, for  $n$ :

```
0 = "Inhospitable"
1 = "Habitable"
2 = "Hospitable"
```

### 3.3.2 Body Type Constants

#### 3.3.2.1 BODY\_TYPE\_ANY

```
BODY_TYPE_ANY = -1
```

#### 3.3.2.2 BODY\_TYPE\_ASTEROIDBELT

```
BODY_TYPE_ASTEROIDBELT = 102
```

#### 3.3.2.3 BODY\_TYPE\_BLACKHOLE

```
BODY_TYPE_BLACKHOLE = 3
```

#### 3.3.2.4 BODY\_TYPE\_BLUEGIANT

```
BODY_TYPE_BLUEGIANT = 10
```

#### 3.3.2.5 BODY\_TYPE\_BROWNDWARF

```
BODY_TYPE_BROWNDWARF = 11
```

**3.3.2.6 BODY\_TYPE\_CLOSEMULT**

```
BODY_TYPE_CLOSEMULT = 12
```

**3.3.2.7 BODY\_TYPE\_DSO**

```
BODY_TYPE_DSO = 13
```

**3.3.2.8 BODY\_TYPE\_FLEET**

```
BODY_TYPE_FLEET = 200
```

**3.3.2.9 BODY\_TYPE\_FREEOBJECT**

```
BODY_TYPE_FREEOBJECT = 5
```

**3.3.2.10 BODY\_TYPE\_GASGIANT**

```
BODY_TYPE_GASGIANT = 101
```

**3.3.2.11 BODY\_TYPE\_MARKER**

```
BODY_TYPE_MARKER = 202
```

**3.3.2.12 BODY\_TYPE\_MEGASTRUCTURE**

```
BODY_TYPE_MEGASTRUCTURE = 107
```

**3.3.2.13 BODY\_TYPE\_MULT**

```
BODY_TYPE_MULT = 2
```

**3.3.2.14 BODY\_TYPE\_NEBULA**

```
BODY_TYPE_NEBULA = 7
```

**3.3.2.15 BODY\_TYPE\_NEUTRON**

```
BODY_TYPE_NEUTRON = 6
```

**3.3.2.16 BODY\_TYPE\_NOTASSIGNED**

```
BODY_TYPE_NOTASSIGNED = 0
```

**3.3.2.17 BODY\_TYPE\_PLANETOID**

```
BODY_TYPE_PLANETOID = 103
```



### 3.3.2.18 BODY\_TYPE\_PLANETSYSTEM

```
BODY_TYPE_PLANETSYSTEM = 105
```

### 3.3.2.19 BODY\_TYPE\_REDGIANT

```
BODY_TYPE_REDGIANT = 9
```

### 3.3.2.20 BODY\_TYPE\_RING

```
BODY_TYPE_RING = 104
```

### 3.3.2.21 BODY\_TYPE\_SHIP

```
BODY_TYPE_SHIP = 201
```

### 3.3.2.22 BODY\_TYPE\_SMALLBODY

```
BODY_TYPE_SMALLBODY = 106
```

### 3.3.2.23 BODY\_TYPE\_STAR

```
BODY_TYPE_STAR = 1
```

### 3.3.2.24 BODY\_TYPE\_STATION

```
BODY_TYPE_STATION = 4
```

### 3.3.2.25 BODY\_TYPE\_TERRESTRIAL

```
BODY_TYPE_TERRESTRIAL = 100
```

### 3.3.2.26 BODY\_TYPE\_WHITEDWARF

```
BODY_TYPE_WHITEDWARF = 8
```

## 3.3.3 Properties

### 3.3.3.1 Albedo

```
Albedo: Float
```

The body's Albedo value. Should range from 0.0 to 1.0

### 3.3.3.2 AngleAscendingNode

```
AngleAscendingNode: Float
```

Should range from 0.0 to 360.0

### 3.3.3.3 AnglePeriapsis

```
AnglePeriapsis: Float
```

Should range from 0.0 to 360.0

### 3.3.3.4 Atmosphere

```
Atmosphere: Float
```

The thickness of the body's atmosphere. Float value equal to or greater than 0. Value is Earth relative, where the density of Earth's atmosphere is 1.0.

### 3.3.3.5 AtmosphereNotes

```
AtmosphereNotes: String
```

### 3.3.3.6 AxialTilt

```
AxialTilt: Float
```

Angle of Axial Tilt. Should be from 0.0 to 360.

### 3.3.3.7 BlazonDisplay

```
BlazonDisplay: Boolean
```

### 3.3.3.8 BlazonImageFile

```
BlazonImageFile: String
```

### 3.3.3.9 BlazonWrap

```
BlazonWrap: Boolean
```

**3.3.3.10 BodyCount**

```
BodyCount: Integer
```

**3.3.3.11 BodyType**

```
BodyType: String
```

**3.3.3.12 ChildHabitability**

```
ChildHabitability: Integer
```

**3.3.3.13 ChildPopulation**

```
ChildPopulation: Integer
```

**3.3.3.14 Color**

```
Color: Integer
```

Color of the star when displayed on the 3D map. Color is stored as an integer as returned by the RGB () function.

**3.3.3.15 Density**

```
Density: Float
```

The body's physical density, relative to Earth (1.0)

**3.3.3.16 Distance**

```
Distance: Float
```

Distance from parent body, in kilometers.

**3.3.3.17 DSOIntensity**

```
DSOIntensity: Integer
```

The DSO display intensity. Should range from 0 to 100.

**3.3.3.18 Eccentricity**

```
Eccentricity: Float
```

The body's orbital Eccentricity. Should range between 0 and 1.0.

### 3.3.3.19 FontStyle

```
FontStyle: Integer
```

The font style used in the 3D display. Constants and their values are:

```
LABEL_FONT_NORMAL = 0
```

```
LABEL_FONT_BOLD = 1
```

```
LABEL_FONT_UNDERLINE = 2
```

```
LABEL_FONT_BOLDUNDERLINE = 3
```

### 3.3.3.20 GMNotes

```
GMNotes: String
```

### 3.3.3.21 Habitability

```
Habitability: Integer
```

Indicates the body's habitability. Constants and their values:

```
HAB_INHOSPITABLE = 0
```

```
HAB_HABITABLE = 1
```

```
HAB_HOSPITABLE = 2
```

### 3.3.3.22 Inclination

```
Inclination: Float
```

Body's orbital inclination. Should range from 0.0 to 360.0.

### 3.3.3.23 LabelColor

```
LabelColor: integer
```

Color of system labels as displayed on the 3D map or in 2D exported graphics. Value can be created using the RGB() function.

### 3.3.3.24 LabelDistance

```
LabelDistance: Float
```

Distance from the camera within which a body's label is displayed on the 3D map.

### 3.3.3.25 Luminosity

```
Luminosity: Float
```

The body's stellar luminosity. Usually only applies to stars. Value is relative to Sol, who's luminosity is 1.0.

### 3.3.3.26 Mass

```
Mass: Float
```

The body's mass. For stars, this value is relative to Sol (which is 1.0). For other bodies (planets) this value is relative to Earth (1.0). A star of mass 2.5, for example, is 2.5 times larger in mass than Sol.

### 3.3.3.27 Name

```
Name: String
```

The body's name.

### 3.3.3.28 Notes

```
Notes: String
```

### 3.3.3.29 Parent

```
Parent: Body
```

The parent body of the body. This may be NULL, or contain another Body object.

### 3.3.3.30 PoliticalAffiliation

```
PoliticalAffiliation: String
```

### 3.3.3.31 Population

```
Population: Float
```

### 3.3.3.32 PreviewImageFile

```
PreviewImageFile: String
```

### 3.3.3.33 Radius

```
Radius: Float
```

For stars, this is the star's radius relative to Sol. For other bodies, this is their radius, in kilometers.

#### 3.3.3.34 RandomSeed

```
RandomSeed: Integer
```

#### 3.3.3.35 RetrogradeOrbit

```
RetrogradeOrbit: Boolean
```

If true, the body orbits its parent in a retrograde fashion.

#### 3.3.3.36 Rotation

```
Rotation: Float
```

The body's rotational period (day length), in hours.

#### 3.3.3.37 RouteDistance

```
RouteDistance: Float
```

The distance from the camera, in light years, within which any routes attached to the body are displayed on the 3D map.

#### 3.3.3.38 Selected

```
Selected: Boolean
```

If True, the body is selected by the user on the 3D map.

#### 3.3.3.39 Spectral

```
Spectral: String
```

This is the spectral class of the star. For example, "G2".

#### 3.3.3.40 SphereInfluence

```
SphereInfluence: Integer
```

For root bodies only, this is the type of Sphere of Influence displayed around a system. Constants and Values are:

SPHERE\_INFLUENCE\_NONE = 0

SPHERE\_INFLUENCE\_SPHERE = 1

SPHERE\_INFLUENCE\_CUBE = 2

### 3.3.3.41 SphereInfluenceColor

```
SphereInfluenceColor: Integer
```

### 3.3.3.42 SphereInfluenceSize

```
SphereInfluenceSize: Float
```

Size, in light years, of the system's Sphere of Influence

### 3.3.3.43 Temp

```
Temp: Float
```

The body's mean surface temperature, in kelvin (0K = -273C). This is the temperature measured from Absolute Zero.

### 3.3.3.44 TimeOffset

```
TimeOffset: Float
```

This is the body's time past periapsis, in days.

### 3.3.3.45 TypeID

```
TypeID: Integer
```

This value determines the body's type. It should be set to one of the [Body Type Constants](#)<sup>[27]</sup>.

### 3.3.3.46 Visible

```
Visible: Boolean
```

Applies to root bodies only. True if the body should be visible on the map, False if it should be hidden (i.e., filtered out)

### 3.3.3.47 Water

```
Water: Float
```

The water index for a body. This should range from 0 to 1.0. 0 means there is no water on a body, 1.0 means the body is completely saturated in water. Depending on the planet's temperature and other factors, a 1.0 may mean the planet has no land surface area (a 'water world'), is covered completely in ice, or is a giant steam bath.

### 3.3.3.48 X

```
X: Float
```

Applies to root bodies only. The body's X position coordinate on the map.

### 3.3.3.49 Y

```
Y: Float
```

Applies to root bodies only. The body's Y position coordinate on the map.

### 3.3.3.50 Z

```
Z: Float
```

Applies to root bodies only. The body's Z position coordinate on the map.

## 3.3.4 Methods

### 3.3.4.1 AddChild

```
AddChild( b: Body): Integer
```

Adds *b* as a child to the body. This adds *b* as the last body in the list of children. To then sort the bodies based on distance, use [SortChildrenByDistance\(\)](#)<sup>[42]</sup>

### 3.3.4.2 ChildrenCount

```
ChildrenCount(): Integer
```

Returns the number of direct children the body has. This does not include any children of direct children of the body - only the direct children.

### 3.3.4.3 ClearChildrenNames

```
ClearChildrenNames()
```

Clears the names for all of the body's children.

### 3.3.4.4 ClearFields

```
ClearFields()
```

Clears all custom fields for the body.

### 3.3.4.5 CopyTo

```
CopyTo( b: Body)
```

Copies the body's properties and children to *b*.

### 3.3.4.6 DefaultTextureFile

```
DefaultTextureFile()
```

Returns the file name of a default preview image file for the body.



### 3.3.4.7 DeleteChildren

```
DeleteChildren()
```

Deletes all children from the body.

### 3.3.4.8 DistanceFromParentStar

```
DistanceFromParentStar(): Float
```

Returns the distance in KM of the body from its parent star. Useful for calculating the distance from a moon to a star.

### 3.3.4.9 FindChildIndex

```
FindChildIndex( b: Body): integer
```

Finds the index of body *b* in the list of the body's direct children, or returns -1 if *b* is not a direct child of the body.

### 3.3.4.10 FormatChildPopulation

```
FormatChildPopulation(): String
```

Returns a reader friendly version of the body's child population (total population of all bodies beneath the body).

### 3.3.4.11 FormatPopulation

```
FormatPopulation(): String
```

Returns a reader friendly version of the body's population.

### 3.3.4.12 FormatSystemPopulation

```
FormatSystemPopulation(): String
```

Returns a reader friendly version of the total system population.

### 3.3.4.13 FormatYearLength

```
FormatYearLength(): String
```

Returns a human-friendly version of the body's orbital period.

### 3.3.4.14 FWEFileName

```
FWEFileName(): string
```

Returns the name of the FWE file that would be used if one was created for this body.

**3.3.4.15 FWEUpdateTerrain**

```
FWEUpdateTerrain()
```

Updates a terrestrial planet's surface map from it's FWE file, if such a file exists.

**3.3.4.16 GetAtmElement**

```
GetAtmElement( n: Integer): String
```

Returns the name of the gas at position *n* in the list of atmospheric components.

**3.3.4.17 GetAtmElementCount**

```
GetAtmElementCount(): Integer
```

Returns the number of atmospheric components.

**3.3.4.18 GetAtmElementPercent**

```
GetAtmElementPercent( n: Integer): Float
```

Returns the atmospheric percentage of the gas at index *n* in the list of atmospheric components.

**3.3.4.19 GetBodyCount**

```
GetBodyCount(): Integer
```

Returns one plus the total number of child bodies beneath the body.

**3.3.4.20 GetChild**

```
GetChild( n: Integer): Body
```

Returns the child body at index *n* in the list of child bodies.

**3.3.4.21 GetChildPopulation**

```
GetChildPopulation(): Float
```

Returns the total population of the body and all child bodies beneath it.

**3.3.4.22 GetField**

```
GetField( sName: String): String
```

Returns the value of the custom field name *sName*.

**3.3.4.23 GetFieldByIndex**

```
GetFieldByIndex( n: Integer): String
```

Gets the field value of the custom field at index *n*.

#### 3.3.4.24 GetFieldCount

```
GetFieldCount(): Integer
```

Returns the number of custom fields stored for this body.

#### 3.3.4.25 GetFieldIndex

```
GetFieldIndex( sName: String): Integer
```

Returns the index of the custom field *sName* in the list of custom fields.

#### 3.3.4.26 GetFieldNameByIndex

```
GetFieldNameByIndex( n: Integer): String
```

Returns the name of the custom field at index *n* in the list of custom fields.

#### 3.3.4.27 GetFWECOLORGroup

```
GetFWECOLORGroup(): FWECOLORGroup
```

Returns the FWECOLORGroup object for the body, if one exists. Otherwise returns Null.

#### 3.3.4.28 GetGeostationary

```
GetGeostationary(): Float
```

Returns the distance in KM of the body's geostationary orbit, if there is a valid one. This method returns 0 if a geostationary orbit does not apply (no rotation), or a -1 if the geostationary orbit lies outside the body's Hill Sphere.

#### 3.3.4.29 GetHillSphere

```
GetHillSphere(): Float
```

Calculates a planet's Hill Sphere - the theoretical farthest orbit at which it can support a moon.

#### 3.3.4.30 GetMaxChildHabitability

```
GetMaxChildHabitability(): Integer
```

Returns the highest habitability rating found on the body or any of the child bodies below it.

#### 3.3.4.31 GetOrbit

```
GetOrbit(): Orbit
```

Returns an Orbit object for the body.

### 3.3.4.32 GetOrbitalParent

```
GetOrbitalParent(): Body
```

Returns the parent of the body being orbited. For example, calling this for a moon will return the parent star.

### 3.3.4.33 GetParentBody

```
GetParentBody(): Body
```

Returns the parent body of the body, or Null if the body is a root body.

### 3.3.4.34 GetParentStar

```
GetParentStar(): Body
```

Returns the parent star for a planet or moon.

### 3.3.4.35 GetRootBody

```
GetRootBody(): Body
```

Returns the root body for the system this body belongs to. The body returned is the body that is displayed on the 3D map. For example, in a single star system, this would return the parent star. This method may return the body itself if it's called for a root body.

### 3.3.4.36 GetSystemPopulation

```
GetSystemPopulation(): Float
```

Returns the total population for the entire system the body is within.

### 3.3.4.37 GreenhouseEffectK

```
GreenhouseEffectK(): Float
```

This returns the difference between the body's mean surface temperature and equilibrium temperature. For planets generated with the program's default system generator, this usually reflects the amount of surface warming due to atmospheric greenhouse effects.

### 3.3.4.38 HasParent

```
HasParent(): Boolean
```

Returns True if the body has a parent body. False if it is a root body and thus does not have a parent.

### 3.3.4.39 InsertBefore

```
InsertBefore( b: Body): Integer
```

Inserts the body b as a child of the body's parent body, placing it into an orbit between the body and

the most previous body in the orbit.

#### 3.3.4.40 ModifySystem

```
ModifySystem()
```

Flags the entire system as modified. Do this after making changes to a body to ensure that the program knows it should save the changes to the sector file.

#### 3.3.4.41 ResetAtmElements

```
ResetAtmElements()
```

Clears all of the body's atmosphere elements from the list of atmospheric components.

#### 3.3.4.42 SetAtmElement

```
SetAtmElement( sGas: String; percent: Float)
```

Adds and sets the atmosphere's percent composition of *sGas* to *percent*.

#### 3.3.4.43 SetChildDesignations

```
SetChildDesignations( s: String)
```

Sets the child names using the default naming scheme (I, II, IV, etc for planets, .1, .2 for moons, etc) for any body in a system that does not already have a name assigned. Parameter *s* is the system name to which suffixes are added.

#### 3.3.4.44 SetField

```
SetField( sName: String, sVal: String)
```

Sets a custom field named *sName* to a value of *sVal*.

#### 3.3.4.45 SetFWECOLORGroup

```
SetFWECOLORGroup( fc: FWECOLORGroup)
```

Applies a FWECOLORGroup object to the body. This FWECOLORGroup object is then used as the color settings for any terrain generated for this body.

#### 3.3.4.46 SetStarMetrics

```
SetStarMetrics()
```

Sets a star's Luminosity and Radius properties from it's Mass property.

#### 3.3.4.47 SetSystemDataConfig

```
SetSystemDataConfig( sdc: SystemDataConfig)
```

Applies a SystemDataConfig object to the body. When a SystemDataConfig object is applied to an individual body like this, values in the the System Data tree in the user interface are displayed according to the rules defined for the SystemDataConfig object.

#### 3.3.4.48 SortChildrenByDistance

```
SortChildrenByDistance()
```

Sorts the body's direct children by their distance from the body.

#### 3.3.4.49 SystemModified

```
SystemModified(): Boolean
```

Returns True if the system this body belongs to has been modified.

#### 3.3.4.50 TempAt

```
TempAt( nDistance: Float, nAlbedo: Float): Float
```

Returns the equilibrium temperature, in K, for bodies at *nDistance* KM from the body and with an Albedo of *nAlbedo*. This obviously only applies to Stars or bodies with a non-zero luminosity.

#### 3.3.4.51 TextureExists

```
TextureExists(): Boolean
```

Returns True if a preview image file for this body exists.

#### 3.3.4.52 TextureName

```
TextureName(): String
```

Returns the name of the body's preview image file. Used primarily for non-planetary bodies.

#### 3.3.4.53 TextureNameFromSpectral

```
TextureNameFromSpectral(): String
```

Returns a star's standard preview image file based on its spectral class.

#### 3.3.4.54 TexturePath

```
TexturePath(): String
```

Returns the path texture files are stored in. Of limited use in v2 and alter.

#### 3.3.4.55 UpdateRootBody

```
UpdateRootBody()
```

Updates the system's root body with the system's total population, habitability, and other statistics. Use this after making changes to populations or habitability, or after adding/deleting child bodies to/

from any body in the system.

### 3.3.4.56 YearLength

```
YearLength(): Float
```

Returns the body's year length (or orbital period, if a moon) in days.

## 3.4 Route Class

### 3.4.1 Related Functions

#### 3.4.1.1 CreateRoute

```
CreateRoute(): Route
```

### 3.4.2 Properties

#### 3.4.2.1 Name

```
Name: String
```

#### 3.4.2.2 RouteType

```
RouteType: String
```

#### 3.4.2.3 Red

```
Red: Float
```

#### 3.4.2.4 Green

```
Green: Float
```

#### 3.4.2.5 Blue

```
Blue: Float
```

#### 3.4.2.6 LineWidth

```
LineWidth: Float
```

### 3.4.2.7 LineStyle

```
LineStyle: Integer
```

This describes the type of line to draw. The possible values are:

```
ROUTE_LINE_SOLID = 0;  
ROUTE_LINE_DASH = 1;  
ROUTE_LINE_SHORTDASH = 2;  
ROUTE_LINE_LONGSHORT = 3;  
ROUTE_LINE_LONGSHORTSHORT = 4;
```

### 3.4.2.8 Stipple

```
Stipple: Word
```

## 3.4.3 Methods

### 3.4.3.1 CopyTo

```
CopyTo( r: Route; bWaypoints: Boolean)
```

### 3.4.3.2 AddSystem

```
AddSystem( b: Body)
```

### 3.4.3.3 AddWaypoint

```
AddWaypoint( wp: RouteWaypoint)
```

### 3.4.3.4 SelectObject

```
SelectObject( x: Integer; y: Integer; filter: Integer): Boolean
```

### 3.4.3.5 ClickedOn

```
ClickedOn( x: Integer; y: Integer; filter: Integer): Boolean
```



**3.4.3.6 WaypointCount**

```
WaypointCount(): Integer
```

**3.4.3.7 GetWaypoint**

```
GetWaypoint( n: Integer): RouteWaypoint
```

**3.4.3.8 DeleteWaypoint**

```
DeleteWaypoint( n: Integer)
```

**3.4.3.9 OnRoute**

```
OnRoute( b: Body): Boolean
```

**3.4.3.10 SetField**

```
SetField( sName: String; sVal: String)
```

**3.4.3.11 GetField**

```
GetField( sName: String): String
```

**3.4.3.12 GetFieldIndex**

```
GetFieldIndex( sName: String): Integer
```

**3.4.3.13 GetFieldByIndex**

```
GetFieldByIndex( n: Integer): String
```

**3.4.3.14 GetFieldNameByIndex**

```
GetFieldNameByIndex( n: Integer): String
```

**3.4.3.15 GetFieldCount**

```
GetFieldCount(): Integer
```

### 3.4.3.16 ClearFields

```
ClearFields()
```

## 3.5 RouteWaypoint Class

### 3.5.1 Related Functions

#### 3.5.1.1 CreateWaypoint

```
CreateWaypoint(): RouteWaypoint
```

### 3.5.2 Properties

#### 3.5.2.1 Body

```
Body: Body
```

#### 3.5.2.2 id

```
id: String
```

#### 3.5.2.3 Pos

```
Pos: Point
```

#### 3.5.2.4 X

```
Y: Float
```

#### 3.5.2.5 Y

```
Y: Float
```

#### 3.5.2.6 Z

```
Z: Float
```

### 3.5.3 Methods

There are no methods in the RouteWaypoint Class.

## 3.6 Subsector

### 3.6.1 Related Functions

#### 3.6.1.1 CreateSubsector

```
CreateSubsector(): Subsector
```

### 3.6.2 Properties

#### 3.6.2.1 Name

```
Name: String
```

#### 3.6.2.2 Color

```
Color: Integer
```

#### 3.6.2.3 ShowLabel

```
ShowLabel: Boolean
```

#### 3.6.2.4 Shape

```
Shape: Integer
```

#### 3.6.2.5 Visible

```
Visible: Boolean
```

**3.6.2.6 Center**

```
Center: Point
```

**3.6.2.7 X**

```
X: Float
```

**3.6.2.8 Y**

```
Y: Float
```

**3.6.2.9 Z**

```
Z: Float
```

**3.6.2.10 SizeX**

```
SizeX: Float
```

**3.6.2.11 SizeY**

```
SizeY: Float
```

**3.6.2.12 SizeZ**

```
SizeZ: Float
```

**3.6.2.13 RotateX**

```
RotateX: Float
```

**3.6.2.14 RotateY**

```
RotateY: Float
```

### 3.6.2.15 RotateZ

```
RotateZ: Float
```

### 3.6.2.16 ShowGrid

```
ShowGrid: Boolean
```

### 3.6.2.17 GridX

```
GridX: Float
```

### 3.6.2.18 GridY

```
GridY: Float
```

### 3.6.2.19 GridZ

```
GridZ: Float
```

### 3.6.2.20 GridTick

```
GridTick: Float
```

### 3.6.2.21 TickLabels

```
TickLabels: Boolean
```

## 3.6.3 Methods

### 3.6.3.1 CopyTo

```
CopyTo( sb: Subsector)
```

### 3.6.3.2 inSubsector

```
inSubsector( pt: Point): Boolean
```

### 3.6.3.3 SetField

```
SetField( sName: String; sVal: String)
```

### 3.6.3.4 GetField

```
GetField( sName: String): String
```

### 3.6.3.5 GetFieldIndex

```
GetFieldIndex( sName: String): integer
```

### 3.6.3.6 GetFieldNameByIndex

```
GetFieldNameByIndex( n: Integer): String
```

### 3.6.3.7 GetFieldCount

```
GetFieldCount(): Integer
```

### 3.6.3.8 ClearFields

```
ClearFields()
```

### 3.6.3.9 GetFieldByIndex

```
GetFieldByIndex( n: Integer): String
```

## 3.7 SystemDataConfig Class

SystemDataConfig objects contain information about the fields that are displayed for a body in the System Data display. Normally, the fields that are displayed are done so based on the type of the object, and how the fields are defined for that type of body at the entire sector level. By creating a SystemDataConfig object, defining fields for it, and assigning it to a body, you can override the default field definitions for that specific body.

To alter the fields for a type of body across the entire sector map, assign the fields using the sector properties window.

```
'Simple example that defines a single field for a star.
oSector = GetCurrentSector()

If oSector.SelectedCount > 0 Then
    b = oSector.GetSelected( 0)

    sdc = CreateSystemDataConfig()
    sdc.AddField "The spectral class is", "spectral", "", False

    b.SetSystemDataConfig( sdc)
End If
RefreshScene
```

## 3.7.1 Related Functions

### 3.7.1.1 CreateSystemDataConfig

```
CreateSystemDataConfig(): SystemDataConfig
```

This creates and returns a SystemDataConfig object. This object can then be assigned to a body to configure which fields are displayed in the System Data display.

### 3.7.1.2 Body:SetSystemDataConfig

You can assign a SystemDataConfig object to an individual body to alter how its fields are displayed in the System Data display by using the [SetSystemDataConfig\(\)](#)<sup>[41]</sup> of the Body class method.

## 3.7.2 Properties

There are no properties of the SystemDataConfig class.

## 3.7.3 Methods

### 3.7.3.1 AddField

```
AddField( sDisp: String, sSource: String, sField: String; bHide: Boolean )
```

This adds a field definition to the SystemDataConfig object.

*sDisp* is the text to display in the System Data display as a field name.

*sSource* is the data source.

*sField* is the name of the custom field to use as the data source, if *sSource* is set to 'custom field'.

*bHide* specifies if the field should be displayed or hidden if the value for the field is empty or zero.

### 3.7.3.2 Clear

```
Clear()
```

This clears all fields from the SystemDataConfig object.

### 3.7.3.3 CopyFrom

```
CopyFrom( sdc: SystemDataConfig)
```

This copies applies the fields of the SystemDataConfig object *sdc*, and applies them to the object.

### 3.7.3.4 Count

```
Count()
```

This method returns the number of fields defined for the SystemDataConfig object.

### 3.7.3.5 DeleteField

```
DeleteField( sDisp: string)
```

This deletes the field who's Display Text is set to *sDisp*.

### 3.7.3.6 FindField

```
FindField( sDisp: String): Integer
```

This finds the field with the display text of *sDisp*, and returns its index in the array of fields. The return value is the position of the field in the array of fields (zero based) or -1 if it is not found. You can use the value returned by this function to then move fields up or down the list.

### 3.7.3.7 MoveUpDown

```
MoveUpDown( n: Integer, nDirection: Integer)
```

Use this method to move the field definition in position *n* up or down one. If *nDirection* is negative, the field is moved towards the top of the display list, if it is positive the field is moved towards the bottom of the field list.

## 3.8 SystemDataConfigHolder

The primary purpose of the SystemDataConfigHolder class is to keep track of which Body type is associated with which SystemDataConfig on the map.

### 3.8.1 Methods



### 3.8.1.1 SetConfigForType

```
SetConfigForType( sdc: SystemDataConfig; BodyType: integer)
```

This applies the SystemDataConfig object, *sdc*, to all bodies on the map that have a [TypeID](#)<sup>35</sup> of *BodyType*.

## 3.9 Orbit Class

### 3.9.1 Related Functions

There are no related functions for the Orbit class. To create an Orbit object, use the [GetOrbit\(\)](#)<sup>39</sup> method of the Body class.

### 3.9.2 Properties

#### 3.9.2.1 CurrentTime

```
CurrentTime: X
```

#### 3.9.2.2 PositionX

```
PositionX: Float
```

#### 3.9.2.3 PositionY

```
PositionY: Float
```

#### 3.9.2.4 PositionZ

```
PositionZ: Float
```

### 3.9.3 Methods

#### 3.9.3.1 SetTime

```
SetTime( t: Float)
```

Parameter *t* is the time, in days. When you call this method, the PositionX, PositionY, and PositionZ properties are set to the location of the body in its orbit at time *t*.

## 3.10 Movie Functions

### 3.10.1 StartMovie

```
StartMovie( sFile: String; nWidth: Integer; nHeight: Integer): Boolean
```

### 3.10.2 AddMovieFrame

```
AddMovieFrame(): Boolean
```

### 3.10.3 FinishMovie

```
FinishMovie(): Boolean
```

## 3.11 InspirationPad Class

### 3.11.1 Related Functions

#### 3.11.1.1 CreateIPadTable

```
CreateIPadTable( sFile: String): InspirationPad
```

Loads the .ipt file named *sFile* and returns an InspirationPad object for that generator.

### 3.11.2 Properties

#### 3.11.2.1 LastResultCount

```
LastResultCount: Integer
```

Returns the number of results from the last call to the Go() method. This is usually the number of repetitions passed to Go(), but may be less if the table restricts the number of results.

### 3.11.3 Methods

### 3.11.3.1 AddVariableDef

```
AddVariableDef( sName: String; sVal: String)
```

### 3.11.3.2 AddDefinition

```
AddDefinition( sName: String; sVal: String)
```

### 3.11.3.3 ClearAllVars

```
ClearAllVars()
```

### 3.11.3.4 ShuffleAll

```
ShuffleAll()
```

### 3.11.3.5 Go

```
Go( n: Integer)
```

Example:

```
'Create a new IPad table object by loading the file named sPath
ipad = CreateIPadTable( sPath)

'set a variable to be accessed by the IPad table.
'you only need to add a variable definition if you need to pass
'some value from a script into an IPad table.
ipad.AddDefinition "somevar", "{1d6}"

'run the table 4 times
ipad.Go 4

'loop through and get each individual results
For i = 1 To 4
    s = ipad.GetResult( i-1)
    MsgBox s
Next

FreeObject ipad
```

### 3.11.3.6 GetResult

```
GetResult( n: integer): String
```

## 3.12 Graphical User Interface Classes

### 3.12.1 Dialog Window

#### 3.12.1.1 Related Functions

##### 3.12.1.1.1 NewDialogWindow

```
NewDialogWindow(): DialogWindow
```

#### 3.12.1.2 Properties

##### 3.12.1.2.1 Top

```
Top: Integer
```

##### 3.12.1.2.2 Left

```
Left: Integer
```

##### 3.12.1.2.3 Width

```
Width: Integer
```

##### 3.12.1.2.4 Height

```
Height: Integer
```

##### 3.12.1.2.5 Centered

```
Centered: Boolean
```

If *true*, the window is centered on the screen, regardless of what the Top and Left properties are set to.

##### 3.12.1.2.6 Caption

```
Caption: String
```

#### 3.12.1.3 Methods

## 3.12.1.3.1 AddTextEdit

```
AddTextEdit(): Widget
```

Creates a Text Edit (Text box) widget and places it on the window.

## 3.12.1.3.2 AddLabel

```
AddLabel(): Widget
```

Creates a Label widget and places it on the window.

## 3.12.1.3.3 AddCheckbox

```
AddCheckbox(): Widget
```

Creates a Check Box widget and places it on the window.

## 3.12.1.3.4 AddRadioGroup

```
AddRadioGroup(): Widget
```

Creates a Radio Button Group widget and places it on the window.

## 3.12.1.3.5 AddMemo

```
AddMemo(): Widget
```

Creates a multi-line Text Box widget and places it on the window.

## 3.12.1.3.6 AddListbox

```
AddListbox(): Widget
```

Creates a List Box widget and places it on the window.

## 3.12.1.3.7 AddDropList

```
AddDropList(): Widget
```

Creates a fixed Drop Down List widget and places it on the window.

## 3.12.1.3.8 AddCombo

```
AddCombo(): Widget
```

Creates a Combo Box widget (an editable drop down list) and places it on the window.

## 3.12.1.3.9 AddGroupBox

```
AddGroupBox(): Widget
```

Creates a Group Box widget and places it on the window.

#### 3.12.1.3.10 AddTrackBar

```
AddTrackBar(): Widget
```

Creates a Track Bar widget (also known as a slider) and places it on the window.

#### 3.12.1.3.11 AddImage

```
AddImage(): Widget
```

Creates an Image widget and places it on the window.

#### 3.12.1.3.12 AddFileOpen

```
AddFileOpen(): Widget
```

Creates a File Open widget and places it on the window. This widget is a combination of a Text Box and a button that lets the user select a file to open.

#### 3.12.1.3.13 AddFileSave

```
AddFileSave(): Widget
```

Creates a File Save widget and places it on the window. This widget is a combination of a Text Box and a button that lets the user select a file to save as.

#### 3.12.1.3.14 AddColorSelector

```
AddColorSelector(): Widget
```

Creates a Color Selector widget and adds it to the window. A Color Selector widget is a colored button that when clicked on displays a color selection window.

#### 3.12.1.3.15 ShowModal

```
ShowModal(): Boolean
```

Displays the DialogWindow object. Returns *true* if the Ok button is pressed, *false* if the Cancel button is pressed.

## 3.12.2 Widget

### 3.12.2.1 Properties

## 3.12.2.1.1 Top

```
Top: Integer
```

## 3.12.2.1.2 Left

```
Left: Integer
```

## 3.12.2.1.3 Width

```
Width: Integer
```

## 3.12.2.1.4 Height

```
Height: Integer
```

## 3.12.2.1.5 Text

```
Text: String
```

This is the contents of a Text Edit or Memo's text box. This is different from the *Caption*, which is the text used to label certain widgets.

## 3.12.2.1.6 Caption

```
Caption: String
```

This is the label for a widget. Not to be confused with a widget's *Text* property which holds a user editable text value.

## 3.12.2.1.7 Checked

```
Checked: Boolean
```

For Checkbox widgets, specifies if the Checkbox is checked or not.

## 3.12.2.1.8 Columns

```
Columns: Integer
```

For Radio Groups, this is the number of columns to use when displaying options.

## 3.12.2.1.9 Color

```
Color: Integer
```

## 3.12.2.1.10 FileMask

```
FileMask: String
```

For FileOpen and FileSave widgets, this is the file mask that's used to filter file names. This is stored as a pipe-delimited string, as such: "description|mask|description|mask".

Example:

```
'To offer filters for .sector files, or files of any type:  
fo.FileMask = "Star Sectors (*.sector)|*.sector|All Files (*.*)|*.*"
```

#### 3.12.2.1.11 FileName

```
FileName: String
```

#### 3.12.2.1.12 MinValue

```
MinValue: Integer
```

This is the minimum value of a TrackBar widget.

#### 3.12.2.1.13 MaxValue

```
MaxValue: Integer
```

This is the maximum value of a TrackBar widget.

#### 3.12.2.1.14 Position

```
Position: Integer
```

This is the current position of a TrackBar widget as set by the user.

#### 3.12.2.1.15 ImageFileName

```
ImageFileName: String
```

#### 3.12.2.1.16 Fit

```
Fit: Boolean
```

Applies to Image widgets. If *true*, the image is scaled to fit the size of the widget. If *false*, it is not (and is cropped if it is too big).

### 3.12.2.2 Methods

#### 3.12.2.2.1 AddItem

```
AddItem(s: String)
```



Adds an item to a ListBox, DropList, or ComboBox type widget.

#### 3.12.2.2.2 SetPosition

```
SetPosition( x: Integer; y: Integer; w: Integer; h: Integer)
```

Sets the widget's position on the window to a Left value of *x*, Top value of *y*, Width of *w*, and Height of *h*.

## 3.13 GenerateOptions Class

### 3.13.1 Properties

#### 3.13.1.1 BodyType

```
BodyType: String
```

This is the textual description of the type of system the user is requesting that the plug-in create. For example, 'Star, Yellow' or 'Multiple Star System'.

#### 3.13.1.2 GenContents

```
GenContents: Boolean
```

Specifies whether or not the user is requesting the plug-in create just a root system body, or generate all child objects (planets, moons) as well.

#### 3.13.1.3 GenPopulations

```
GenPopulations: Boolean
```

Specifies if the user wants the generator to generate planet populations and space stations (true) or not (false).

#### 3.13.1.4 NameGenerator

```
NameGenerator: String
```

The name of the name generator the user selected.

### 3.13.2 Methods

There are no methods for this class.

## 3.14 FWIColorGroup

The FWIColorGroup class contains color settings and other terrain generation options that are passed to the Fractal World Explorer terrain generation engine.

### 3.14.1 Properties

#### 3.14.1.1 AtmColor

```
AtmColor: Integer
```

This is the color of the Atmosphere.

#### 3.14.1.2 Beach

```
Beach: Integer
```

Color of shoreline areas, where elevation < 5m

#### 3.14.1.3 DeepWater

```
DeepWater: Integer
```

Water color where sea depth exceeds 100m.

#### 3.14.1.4 Dry

```
Dry: Integer
```

The terrain color in absolute dry conditions (0 precipitation). This color is combined in a gradient with the land color (based on the LowLand and HighLand properties) in dry areas.

#### 3.14.1.5 GenAtm

```
GenAtm: Boolean
```

If True, the FWE engine generates an atmosphere. If False, it will not generate an atmosphere.

#### 3.14.1.6 HighLand

```
HighLand: Integer
```

The color of land at the map's maximum altitude. Gradient colors between LowLand and HighLand are used when calculating terrain colors at specific elevation above ground.

#### 3.14.1.7 Ice

```
Ice: Integer
```

The color of ice caps on the planet.

### 3.14.1.8 LowLand

```
LowLand: Integer
```

The color of land at 5m. Gradient colors between LowLand and HighLand are used when calculating terrain colors at specific elevation above ground.

### 3.14.1.9 LowWater

```
LowWater: Integer
```

Water color where sea depth is less than 100m.

## 3.15 Utility Functions

### 3.15.1 AppVersionBuild

```
AppVersionBuild(): String
```

Returns the full version & build information of the application. This includes both major, minor, and build values.

### 3.15.2 AppVersionMajor

```
AppVersionMajor(): String
```

Returns the major version number of the application. For AstroSynthesis v2.01, this would be '2.00'.

### 3.15.3 AppVersionMinor

```
AppVersionMinor(): String
```

Returns the major and minor version numbers of the application. This includes both the major version number (the number to the left of the decimal) and the minor version numbers (numbers to the right of the decimal). For AstroSynthesis v2.01, this would be '2.01'

### 3.15.4 AstroDirectory

```
AstroDirectory(): String
```

Returns the directory in which the main AstroSynthesis executable resides. This is typically 'c:\Program Files\nbos\Astro'.

### 3.15.5 ClearMap

```
ClearMap()
```

This function clears the current map as if the user had selected File - New from the menu.

### 3.15.6 DeselectEditingBody

```
DeselectEditingBody()
```

If a body is selected in the System Data tree, this function de-selects it.

### 3.15.7 EditBody

```
EditBody( b: Body)
```

This displays the Edit Properties window for the the body, *b*, and allows the user to make changes to the body's values.

### 3.15.8 EditingBody

```
EditingBody(): Body
```

Returns the body currently selected in the System Data tree. This function is most useful for creating [pop-up menu](#)<sup>[5]</sup> scripts.

### 3.15.9 FileHandlerFileName

```
FileHandlerFileName(): String
```

Used for creating [file handlers](#)<sup>[3]</sup>. This function returns the name of the file the user is attempting to open.

### 3.15.10 FreeObject

```
FreeObject( o: Object)
```

Any object created with a Create\*() function that is not added into a map (such as a system or route) should be freed by sending it into this function. For example, if you create a body that you do not use in your map, free the memory it is using it by using this function when you no longer need to reference the object. Do not free objects that you add into a map, such as bodies or routes that you intend to save.

### 3.15.11 GetGenerateOptions

```
GetGenerateOptions(): GenerateOptions
```

Returns an object containing information about what system generation options have been selected by the user when creating a *single* system. Use this when creating a [body generator](#)<sup>[2]</sup> plug-in.

### 3.15.12 GetKey

```
GetKey(): Integer
```

Returns an integer representing the first key found in the message queue, or -1 if no key has been pressed. Use this function to check for the Escape key being pressed (27), or even control sector navigation by responding to arrow key presses.

### 3.15.13 isNull

```
isNull(o: Object)
```

Reports if the object *o* is Null, or not pointing to a valid object. Attempting to access a null object's properties or methods will likely result in an access violation, so be sure to use this to test the return value of any function or method that might return a null object (such as [EditingBody\(\)](#)<sup>[64]</sup>).

### 3.15.14 Pause

```
Pause(n: Integer)
```

Pauses the execution of the script for *n* milliseconds.

### 3.15.15 PluginDirectory

```
PluginDirectory(): String
```

Returns the Plugins directory for AstroSynthesis. This is the directory where AstroSynthesis is installed plus 'Plugins'.

### 3.15.16 ProcessMessages

```
ProcessMessages()
```

Processes Windows application messages waiting in queue. Call this function just prior to calling processor intensive Alien API calls to allow the application to process Paint (WM\_PAINT) and other messages so that the application does not appear to hang. For example, call this function prior to calling methods to create surface maps.

### 3.15.17 RefreshScene

```
RefreshScene()
```

Refreshes the 3D display. Call this to update the display with new systems, positions, or [rendering messages](#)<sup>[11]</sup>.

### 3.15.18 ShowSystemDiagram

```
ShowSystemDiagram( b: Body)
```

This displays the system diagram (solar system/moon system view) for the body *b*. Only call this for Gas Giants, Terrestrial planets, and Stars.

### 3.15.19 UpdateCaption

```
UpdateCaption()
```

Re-calculates system statistics and updates the program window's caption. Use this after you change

the number of systems or alter filters in code.

## 3.16 AstroSearcher Class

The AstroSearcher Class encapsulates AstroSynthesis' Natural Language Search capabilities. You can use the AstroSearcher Class to execute Natural Language Search queries in script, and cycle through the matched bodies.

```
' This example finds the root bodies of all systems
' with a hospitable world and sets their label color to purple.

sector = GetCurrentSector()

srch = NewAstroSearcher( sector)
srch.Search "systems with hospitable"
n = srch.GetResultCount()
For i = 1 To n
    o = srch.GetResult( i)
    o.LabelColor = RGB( 255, 0, 255)
Next

DeleteAstroSearcher srch
RefreshScene
```

### 3.16.1 Related Functions

#### 3.16.1.1 CreateAstroSearcher

```
CreateAstroSearcher( o: Sector): Searcher
```

This function creates an new AstroSearcher object for the given sector.

When you are finished running your query, use [FreeObject](#)<sup>64</sup>() to free the AstroSearcher object

### 3.16.2 Properties

#### 3.16.2.1 ErrorMsg

```
Error: string
```

This property contains any error messages that are generated due to problems parsing a passed Natrual Language Search query.

### 3.16.3 Methods

#### 3.16.3.1 Search

```
Search(sNLS: String)
```

This method performs a query using Natural Language Search. *sNLS* is a string containing an AstroSynthesis Natural Language Search query. If the query cannot be parsed, the Error property will contain information about the error.

Note that the number of results returned is not limited for searches conducted by scripts. So take care with queries used when running scripts, and use the 'First' command to limit results if needed.

#### 3.16.3.2 GetResultCount

```
GetResultCount(): integer
```

This function returns the number of bodies found by a search. Use this in conjunction with GetResult() to access the search results.

#### 3.16.3.3 GetResult

```
GetResult( nIndex: integer): Body
```

This function returns the *n*-th body in the list of results for a search. Use in conjunction with GetResultCount() to cycle through all the results of a query. The parameter *nIndex* is 1 based. So to obtain the first result, pass 1, not 0.