Introduction

The GLLib main function is to be the hearth of any game creation at Gameloft. The GLLib provides the developer with methods to handle game key events by doing the mapping to your device. Package management for your resources is also provided by the GLLib class. You have access also to basic math support, IGP, Xplayer ...
Support

To have support contact world-glib
GLLib Modules

Here is a list of all modules:

- **ASprite**
- **GLLib**
  - **GLLib : Core System.**
  - **GLLib : Keypad System**
    - Vars for Key Accumulation
    - Vars without Key Accumulation
    - GLKey
  - **GLLib : Debug Utils**
  - **GLLib : Math Utils**
  - **GLLib : File/Package System**
  - **GLLib : MIDP wrapper**
  - **GLLib : memory**
  - **GLLib : text**
    - **GLLang**
  - **GLLib : Recordstore management**
    - **GLLib : GLLib Profiler**
  - **GLLib : GLLib Profiler**
- **GLLib Configuration file**
- **GLLibPathFinding**
- **GLLibPlayer**
  - **GLLibPlayer : AnimationPlayer**
  - **GLLibPlayer : SoundPlayer**
  - **GLLibPlayer : Tileset**
- **XPlayer**
  - **HTTP (used by XPlayer and License)**
  - **TCP (used by XPlayer)**
  - **XPlayer core**
  - **Xplayer Error Code**
ASprite
## Packages

| package | javax.microedition.lcdui |
### Classes

<table>
<thead>
<tr>
<th>class</th>
<th>ASprite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASprite</td>
</tr>
</tbody>
</table>

*ASprite* Aurora Sprite class. [More...](#)

---

*Generated on Tue Sep 23 23:50:30 2008 for GLib by doxygen 1.5.2*
GLLib
<table>
<thead>
<tr>
<th>package</th>
<th>java.io</th>
</tr>
</thead>
<tbody>
<tr>
<td>package</td>
<td>java.util</td>
</tr>
<tr>
<td>package</td>
<td>javax.microedition.rms</td>
</tr>
</tbody>
</table>
## Modules

<table>
<thead>
<tr>
<th>GLLib: Core System.</th>
<th>all low level phone integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLLib: Keypad System</td>
<td>key input method</td>
</tr>
<tr>
<td>GLLib: Debug Utils</td>
<td>debugging function to log some message (only in debug version of the lib)</td>
</tr>
<tr>
<td>GLLib: Math Utils</td>
<td>mathematic function</td>
</tr>
<tr>
<td>GLLib: File/Package System</td>
<td>handle data file access (read only)</td>
</tr>
<tr>
<td>GLLib: MIDP wrapper</td>
<td>wrapper for some usual midp function, use this instead of the midp ones</td>
</tr>
<tr>
<td>GLLib: memory</td>
<td>memory access function (byte array manipulation)</td>
</tr>
<tr>
<td>GLLib: text</td>
<td>text decoding and caching function</td>
</tr>
<tr>
<td>GLLib: Recordstore management</td>
<td>handle RMS (save data) access (read write)</td>
</tr>
<tr>
<td>GLLib: GLLib Profiler</td>
<td>Utility functions for profiling in emulator and phone.</td>
</tr>
</tbody>
</table>
### Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLib</th>
</tr>
</thead>
</table>

*The GLLib class is the main class to do a game creation at Gameloft.* [More...](#)

---

*Generated on Tue Sep 23 23:05:30 2008 for GLLib by doxygen 1.5.2*
GLLib : Core System.
[GLLib]

all low level phone integration More...
### Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void CheckAndDumpConfig ()</td>
<td></td>
</tr>
<tr>
<td>void Game_Run () throws Exception</td>
<td>Function to be implemented in every game.</td>
</tr>
<tr>
<td>abstract void Game_update () throws Exception</td>
<td>Function to be implemented in every game.</td>
</tr>
<tr>
<td>Image GetSoftwareDoubleBuffer ()</td>
<td></td>
</tr>
<tr>
<td>void hideNotify ()</td>
<td>Override of Canvas.hideNotify().</td>
</tr>
<tr>
<td>void Init ()</td>
<td>Initialize and start the game engine.</td>
</tr>
<tr>
<td>void paint (Graphics _g)</td>
<td>Standard rendering function.</td>
</tr>
<tr>
<td>void Pause ()</td>
<td>Pause the game engine.</td>
</tr>
<tr>
<td>void Quit ()</td>
<td>Request to quit the game engine.</td>
</tr>
<tr>
<td>void Resume ()</td>
<td>Resume the game engine.</td>
</tr>
<tr>
<td>void run ()</td>
<td>Game engine main loop/thread.</td>
</tr>
<tr>
<td>static void SetupDefaultKey ()</td>
<td>Setup the default key association for this device.</td>
</tr>
<tr>
<td>void SetupDisplay ()</td>
<td>Setup the display.</td>
</tr>
<tr>
<td>void showNotify ()</td>
<td>Override of Canvas.showNotify().</td>
</tr>
<tr>
<td>void sizeChanged (int w, int h)</td>
<td>Called when the drawable area has been changed.</td>
</tr>
<tr>
<td>void UnInit ()</td>
<td>Quit game engine, free all memory and object.</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <code>javax.microedition.lcdui.Graphics</code></td>
<td><code>g = null</code></td>
<td>Graphics context where all rendering operation will happen.</td>
</tr>
<tr>
<td>static <code>javax.microedition.midlet.MIDlet</code></td>
<td><code>s_application</code></td>
<td>Reference to the application. Usually its a reference to a midlet (or <code>IApplication</code> for doja).</td>
</tr>
<tr>
<td>static <code>Display</code></td>
<td><code>s_display</code></td>
<td>Reference to the display</td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>s_game_currentFrameNB</code></td>
<td>Current frame number. Increased every frame.</td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>s_game_FPSAverage</code></td>
<td>Average fps * 100.</td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>s_game_frameDT</code></td>
<td>Delta time between the previous frame and this one.</td>
</tr>
<tr>
<td>static <code>boolean</code></td>
<td><code>s_game_interruptNotify</code></td>
<td>Interrupt notifier. Set to true when an interrupt occurred.</td>
</tr>
<tr>
<td>static <code>boolean</code></td>
<td><code>s_game_isPaused</code></td>
<td>Pause state of the game. True if the game is paused.</td>
</tr>
<tr>
<td>static <code>long</code></td>
<td><code>s_game_lastFrameTime</code></td>
<td>Similar to <code>s_game_timeWhenFrameStart</code> but internally used for <code>GLLibConfig.useFakeInterruptHandling</code>.</td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>s_game_state</code></td>
<td>Current game state.</td>
</tr>
<tr>
<td>static <code>long</code></td>
<td><code>s_game_timeWhenFrameStart</code></td>
<td>Current time when the frame started.</td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>s_game_totalExecutionTime</code></td>
<td>Total game execution time.</td>
</tr>
<tr>
<td>static <code>GLLib</code></td>
<td><code>s_gllib_instance</code></td>
<td>The only instance of this <code>GLLib</code> class.</td>
</tr>
</tbody>
</table>
Detailed Description

all low level phone integration
### Function Documentation

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>void CheckAndDumpConfig()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>void Game_Run() throws Exception</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function to be implemented in every game.

This is where you put the code of your game. This function will be called once per frame. You have to do the game specific run code from within. The method is not abstract because overriding it should not be mandatory and should not affect existing projects. This function is called from the run call of the main thread.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract void Game_update() throws Exception</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function to be implemented in every game.

This is where you put the code of your game. This function will be called once per frame. You have to do the game logic/AI and painting from within. This function is called from the paint call of this canvas.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image GetSoftwareDoubleBuffer()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics GetSoftwareDoubleBufferGraphics()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Package</th>
<th>Inherited</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLLib(Object application, Object display)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Constructor.
The constructor, will keep reference to the application and display, it will also setup its display This call wont start the engine of the game, you need to call init to start the game engine.

**Parameters:**

- `application` Reference on the application (midlet, IApplication, etc...).
- `display` Reference on the display.

**See also:**

- GLLib.init

---

```java
void hideNotify() [inherited]
```

Override of Canvas.hideNotify().

Standard devices will call this function when the game is interrupted.

**See also:**

- GLLib.pause

---

```java
void Init() [protected, inherited]
```

Initialize and start the game engine.

Allocate basic game variable/structure, launch game thread.

---

```java
void paint(Graphics _g) [inherited]
```

Standard rendering function.

This function is called by the device to enable us to draw on the display. Override of Canvas.paint.

**Note:**

The game should not call this function, it will be called automatically by the device, request are made in GLLib.run. The
game drawing code has to go into the abstract function `GLLib.Game_update`.

See also: `GLLib.Game_update`

```java
void Pause( ) [protected, inherited]
```

Pause the game engine.

Can be called by the Midlet, `GLLib.hideNotify` or the game itself.

See also: `GLLib.hideNotify`

```java
void Quit( ) [protected, inherited]
```

Request to quit the game engine.

When the user select exit/quit (or for any other reason the request quit a game) you should use this function.

```java
void Resume( ) [protected, inherited]
```

Resume the game engine.

Can be called by the Midlet or `GLLib.showNotify`. If the game was not in pause state, nothing happens.

See also: `GLLib.showNotify`

```java
void run( ) [inherited]
```

Game engine main loop/thread.
Override of Runnable.run.

**Note:**
The game should not call this function, it will be called automatically by the thread. The game code has to go into the abstract function `GLLib.Game_update`.

**See also:**
`GLLib.Game_update`

---

```java
static void SetupDefaultKey () [static, protected, inherited]
```

Setup the default key association for this device.

in default key configuration, key2=up, key8=down, key4=left, key6=right and key5=fire
if you want to separate those association, you need to reset the key configuration (Game_KeyClearKeyCode)
and use Game_keySetKeyCode so set your key code association

**See also:**
`Game_keySetKeyCode`
`Game_keyClearKeyCode`

---

```java
void SetupDisplay () [protected, inherited]
```

Setup the display.

Does different thing depending on the device.
For MIDP20 it will setup the display to this instance of `GLLib` and put the game in fullscreen.
For NOLIA UI it will setup the display to this instance of `GLLib` and set the command listener to null.
etc...
void showNotify() [inherited]

Override of Canvas.showNotify().

Standard devices will call this function when the game is made visible on the display.

See also:
GLLib.resume

void sizeChanged(int w,
int h)
[inherited]

Called when the drawable area has been changed.

void UnInit() [protected, inherited]

Quit game engine, free all memory and object.

This function will be called automatically at the end of the game loop.
## Variable Documentation

### `javax.microedition.lcdui.Graphics g = null` [static, inherited]

Graphics context where all rendering operation will happen.

This graphic context can be changed/reset with `setCurrentGraphics`.

**See also:**

`setCurrentGraphics`

### `javax.microedition.midlet.MIDlet s_application` [static, package, inherited]

Reference to the application. Usually its a reference to a midlet (or `IApplication` for doja).

### `Display s_display` [static, package, inherited]

Reference to the display

### `int s_game_currentFrameNB` [static, package, inherited]

Current frame number. Increased every frame.

### `int s_game_FPSAverage` [static, package, inherited]

Average fps * 100.

**Note:**

Valid only if `GLLibConfig.useFrameDT` is true.

### `int s_game_frameDT` [static, package, inherited]
Delta time between the previous frame and this one.

Note: valid only if `GLLibConfig.useFrameDT` is true.

`boolean s_game_interruptNotify` [static, package, inherited]

Interrupt notifier. Set to true when an interrupt occurred.

`boolean s_game_isPaused` [static, package, inherited]

Pause state of the game. True if the game is paused.

`long s_game_lastFrameTime` [static, package, inherited]

Similar to `s_game_timeWhenFrameStart` but internally used for `GLLibConfig.useFakeInterruptHandling`.

`int s_game_state` [static, package, inherited]

Current game state.

`long s_game_timeWhenFrameStart` [static, package, inherited]

Current time when the frame started.

`int s_game_totalExecutionTime` [static, package, inherited]

Total game execution time.

Note: valid only if `GLLibConfig.useFrameDT` is true.
The only instance of this **GLLib** class.
GLLib : Keypad System

[GLLib]

key input method More...
## Modules

### Vars for Key Accumulation
*Used only if $\text{GLLibConfig.useKeyAccumulation} == \text{true}$.*

### Vars without Key Accumulation
*Used only if $\text{GLLibConfig.useKeyAccumulation} == \text{false}$.*

## GLKey

### static void `Game_KeyClearKeyCode` ()
*Clear all the key code. No key input will be valid until you define new key association using `Game_keySetKeyCode`.*

### static void `Game_KeySetKeyCode` (boolean gameAction, int keyCode, int key)
*Set the keycode for a specific key.*

### static int `IsAnyKeyDown` ()
*Is there any key that is being held.*

### static boolean `IsKeyDown` (int keyFlag)
*Query a key to know if its down/pressed.*

### static boolean `IsKeyUp` (int keyFlag)
*Query a key to know if its up/unpressed.*

### void `keyPressed` (int keyCode)
*Key pressed method.*

### void `keyReleased` (int keyCode)
*Key released.*

### static void `ResetAKey` (int keyFlag)
*Reset one key to unpressed state.*

### static void `ResetKey` ()
*Reset all keys to unpressed.*

### static int `WasAnyKeyPressed` ()
*Was there any key just pressed.*

### static int `WasAnyKeyReleased` ()
*Was there any key just released.*

### static boolean `WasKeyPressed` (int keyFlag)
*Was there a key just pressed.*

### static boolean `WasKeyReleased` (int keyFlag)
*Was there a key just released.*
Detailed Description

key input method
Function Documentation

```java
static void Game_KeyClearKeyCode() [static, inherited]

clear all the key code. no key input will be valid until you define new
key association using Game_keySetKeyCode

static void Game_KeySetKeyCode(boolean gameAction,
       int keyCode,
       int key)
[static, inherited]

set the keycode for a specific key

Parameters:
    gameAction  set to true if the key code is related to a game action.
    keyCode     keycode to associate.(eg Canvas.LEFT for example)
    key         to associate with this key code. (eg GLKey.k_left for example)

static int IsAnyKeyDown() [static, inherited]

Is there any key that is being held.

Returns:
    GLKey.k_invalid if no key is held, the correct GLKey value
    otherwise.

static boolean IsKeyDown(int keyFlag)
[static, inherited]

Query a key to know if its down/pressed.

Parameters:
    keyFlag  Key to query.

Returns:
```
Boolean value, true if the key is down/pressed.

```java
static boolean IsKeyUp (int keyFlag) [static, inherited]
```

Query a key to know if its up/unpressed.

**Parameters:**

keyFlag  Key to query.

**Returns:**

Boolean value, true if the key is up/unpressed.

```java
void keyPressed (int keyCode) [protected, inherited]
```

Key pressed method.

Called when a key is pressed.

**Parameters:**

keyCode  Key Code from the device. To be translated to GLLib constants.

**See also:**

Canvas.keyPressed

```java
void keyReleased (int keyCode) [protected, inherited]
```

Key released.

Called when a previously pressed key is released.

**Parameters:**

keyCode  Key Code from the device. To be translated to GLLib constants.

**See also:**

Canvas.keyReleased
static void ResetAKey ( int keyFlag ) [static, inherited]

Reset one key to unpressed state.

**Parameters:**

*keyFlag*  Key to reset state.

**Note:**

If `GLLibConfig.useKeyAccumulation` is False, this call is the same as ResetKey.

static void ResetKey ( ) [static, inherited]

Reset all keys to unpressed.

static int WasAnyKeyPressed ( ) [static, inherited]

Was there any key just pressed.

**Returns:**

`GLKey.k_invalid` if no key was pressed, the correct `GLKey` value otherwise.

static int WasAnyKeyReleased ( ) [static, inherited]

Was there any key just released.

**Returns:**

`GLKey.k_invalid` if no key was released, the correct `GLKey` value otherwise.

static boolean WasKeyPressed ( int keyFlag ) [static, inherited]

Was there a key just pressed.
### Parameters:

*keyFlag* Key to query.

### Returns:

Boolean value, true if the key was just pressed.

```java
static boolean WasKeyReleased ( int keyFlag ) [static, inherited]
```

Was there a key just released.

### Parameters:

*keyFlag* Key to query.

### Returns:

Boolean value, true if the key was just released.
Vars for Key Accumulation
[GLLib : Keypad System]

Used only if GLLibConfig.useKeyAccumulation == true. More...
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>s_game_keyEventIndex</td>
<td>Key indice of the key which had an event. -1 if no event.</td>
</tr>
<tr>
<td>static int</td>
<td>s_game_keyJustPressed</td>
<td>Key indice of the key that was just pressed. -1 if no key was just pressed.</td>
</tr>
<tr>
<td>static long</td>
<td>s_game_keyPressedTime</td>
<td>Time of the last key press.</td>
</tr>
<tr>
<td>static int</td>
<td>s_keyLastKeyPressUntranslatedCode = -9999</td>
<td>Untranslated value of the last keycode. Used before passing through Game_TranslateKeyCode.</td>
</tr>
<tr>
<td>static int</td>
<td>s_keyLastKeyStates</td>
<td>Last key state obtained through getKeyStates().</td>
</tr>
<tr>
<td>static byte</td>
<td>s_keyState []</td>
<td>Current key states bufferised.</td>
</tr>
<tr>
<td>static byte</td>
<td>s_keyStateRT []</td>
<td>RealTime key buffer. User should use s_keyState instead.</td>
</tr>
</tbody>
</table>
Detailed Description

Used only if `GLLibConfig.useKeyAccumulation == true`. 
### Variable Documentation

#### `int s_game_keyEventIndex [static, package, inherited]`

Key indice of the key which had an event. -1 if no event.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.

#### `int s_game_keyJustPressed [static, package, inherited]`

Key indice of the key that was just pressed. -1 if no key was just pressed.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.

#### `long s_game_keyPressedTime [static, package, inherited]`

Time of the last key press.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.

#### `int s_keyLastKeyPressUntranslatedCode = -9999 [static, package, inherited]`

Untranslated value of the last keycode. Used before passing through Game_TranslateKeyCode.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.

#### `int s_keyLastKeyStates [static, package, inherited]`
Last key state obtained through getKeyStates().

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true and if using GameCanvas.

```java
byte s_keyState[] [static, package, inherited]
```

Current key states bufferised.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.

```java
byte s_keyStateRT[] [static, package, inherited]
```

RealTime key buffer. User should use `s_keyState` instead.

**Note:**
valid only if `GLLibConfig.useKeyAccumulation` is true.
Vars without Key Accumulation

[GLLib : Keypad System]

Used only if `GLLibConfig.useKeyAccumulation == false`. More...
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>m_current_keys_pressed</td>
<td>Current keys pressed.</td>
</tr>
<tr>
<td>static int</td>
<td>m_current_keys_released</td>
<td>Current keys released.</td>
</tr>
<tr>
<td>static int</td>
<td>m_current_keys_state</td>
<td>Current keys state.</td>
</tr>
<tr>
<td>static int</td>
<td>m_keys_pressed</td>
<td>Previous frame keys pressed.</td>
</tr>
<tr>
<td>static int</td>
<td>m_keys_released</td>
<td>Previous frame keys released.</td>
</tr>
<tr>
<td>static int</td>
<td>m_keys_state</td>
<td>Previous frame keys state.</td>
</tr>
<tr>
<td>static int</td>
<td>m_last_key_pressed</td>
<td>Previous key pressed.</td>
</tr>
</tbody>
</table>

- m_current_keys_pressed
- m_current_keys_released
- m_current_keys_state
- m_keys_pressed
- m_keys_released
- m_keys_state
- m_last_key_pressed = -9999

*Previous key pressed.*
Detailed Description

Used only if `GLibConfig.useKeyAccumulation` == false.
Variable Documentation

**int m_current_keys_pressed** [static, package, inherited]

Current keys pressed.

**Note:**

is valid only if `GLLibConfig.useKeyAccumulation` is false.

**int m_current_keys_released** [static, package, inherited]

Current keys released.

**Note:**

is valid only if `GLLibConfig.useKeyAccumulation` is false.

**int m_current_keys_state** [static, package, inherited]

Current keys state.

**Note:**

is valid only if `GLLibConfig.useKeyAccumulation` is false.

**int m_keys_pressed** [static, package, inherited]

Previous frame keys pressed.

**Note:**

is valid only if `GLLibConfig.useKeyAccumulation` is false.

**int m_keys_released** [static, package, inherited]

Previous frame keys released.
Note:
is valid only if `GLLibConfig.useKeyAccumulation` is false.

```
int m_keys_state [static, package, inherited]
```

Previous frame keys state.

Note:
is valid only if `GLLibConfig.useKeyAccumulation` is false.

```
int m_last_key_pressed = -9999 [static, package, inherited]
```

Previous key pressed.

Note:
is valid only if `GLLibConfig.useKeyAccumulation` is false.
GLKey

[GLLib : Keypad System]
## Classes

<table>
<thead>
<tr>
<th>interface</th>
<th>GLKey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Game key code interface. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLib by [doxygen](https://www.doxygen.org/) 1.5.2
GLLib : Debug Utils

[GLLib]

documenting function to log some message (only in debug version of the lib) More...
## Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Assert</strong> (boolean test, String errMessage)</td>
<td>Assertion test.</td>
</tr>
<tr>
<td><strong>Dbg</strong> (String log)</td>
<td>Log a debug string on the console.</td>
</tr>
<tr>
<td><strong>Print</strong> (String log)</td>
<td>print a string in out stream, does not append character return at the end of line (as opposed to Dbg)</td>
</tr>
<tr>
<td><strong>Warning</strong> (String message)</td>
<td>display a warning</td>
</tr>
</tbody>
</table>
Detailed Description

debbuging function to log some message (only in debug version of the lib)
Function Documentation

static void Assert ( boolean test, String errMessage )
[static, package, inherited]

Assertion test.

Parameters:

  test  Test to assert. Trigger the errMessage if test is false, nothing if true.
  errMessage Error message that should be displayed, if the test is false.

Note:
  Does nothing in RELEASE.

static void Dbg ( String log )  [static, package, inherited]

Log a debug string on the console.

Parameters:

  log  String to print to the console.

Note:
  Does nothing in RELEASE.

static void Print ( String log )  [static, package, inherited]

print a string in out stream, does not append character return at the
end of line (as opposed to Dbg)

Parameters:

  log  String to print to the console.

Note:
  Does nothing in RELEASE.
static void Warning ( String message ) [static, package, inherited]

display a warning

Parameters:

message message that should be displayed

Note:

Does nothing in RELEASE.
GLLib : Math Utils
[GLLib]

mathematic function More...
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConvertFixedPointToString</code></td>
<td>(int value, int precision)</td>
<td>Converts a <strong>GLLib</strong> Fixed Point number to a string in base 10.</td>
</tr>
<tr>
<td><code>Math_Abs</code></td>
<td>(long a)</td>
<td></td>
</tr>
<tr>
<td><code>Math_Abs</code></td>
<td>(int a)</td>
<td></td>
</tr>
<tr>
<td><code>Math_Atan</code></td>
<td>(int dx, int dy)</td>
<td>Arctangent.</td>
</tr>
<tr>
<td><code>Math_AtanSlow</code></td>
<td>(int dx, int dy)</td>
<td>Arctangent, very slow but accurate method, it find angle by dichotomy.</td>
</tr>
<tr>
<td><code>Math_Bezier2D</code></td>
<td>(int x1, int y1, int x2, int y2, int x3, int y3, int interp)</td>
<td>Three control point 2D Bezier interpolation.</td>
</tr>
<tr>
<td><code>Math_Bezier3D</code></td>
<td>(int x1, int y1, int z1, int x2, int y2, int z2, int x3, int y3, int z3, int interp)</td>
<td>Three control point 3D Bezier interpolation.</td>
</tr>
<tr>
<td><code>Math_Cos</code></td>
<td>(int angle)</td>
<td>Cosinus.</td>
</tr>
<tr>
<td><code>Math_DegreeToFixedPointAngle</code></td>
<td>(int a)</td>
<td>Convert a degree angle into a fixed point angle.</td>
</tr>
<tr>
<td><code>Math_Det</code></td>
<td>(int x1, int y1, int x2, int y2)</td>
<td>Compute the determinant of 2 2Dvector.</td>
</tr>
<tr>
<td><code>Math_DistPointLine</code></td>
<td>(int x0, int y0, int x1, int y1, int x2, int y2) throws Exception</td>
<td>Compute the distance from a point to a segment/line.</td>
</tr>
<tr>
<td><code>Math_Div10</code></td>
<td>(int a)</td>
<td>Divide by 10 a number, with result in fixed point.</td>
</tr>
<tr>
<td><code>Math_DotProduct</code></td>
<td>(int x1, int y1, int x2, int y2)</td>
<td>Compute the dot product of 2 2Dvector.</td>
</tr>
<tr>
<td><code>Math_FixedPoint_Add</code></td>
<td>(int summand1, int summand2)</td>
<td>Addition for Fixed Point.</td>
</tr>
<tr>
<td><code>Math_FixedPoint_DegreesToAngleFixedPoint</code></td>
<td>(int angle)</td>
<td>Convert an angle from degrees to fixed point angle.</td>
</tr>
<tr>
<td><code>Math_FixedPoint_DegreesToRadians</code></td>
<td>(int angle)</td>
<td>Convert an angle from degrees to radians (both fixed point and not, since this is just a ratio).</td>
</tr>
<tr>
<td><code>Math_FixedPoint_Det</code></td>
<td>(int x1, int y1, int x2, int y2)</td>
<td>Compute the determinant of 2 2Dvector in Fixed Point.</td>
</tr>
<tr>
<td><code>Math_FixedPoint_Divide</code></td>
<td>(int dividend, int divisor)</td>
<td>Division for Fixed Point.</td>
</tr>
<tr>
<td><code>Math_FixedPoint_DotProduct</code></td>
<td>(int x1, int y1, int x2, int y2)</td>
<td></td>
</tr>
</tbody>
</table>
Compute the dot product of 2 2D vector in Fixed Point.

static int Math_FixedPoint_LineCircleIntersect (int in_x1, int in_y1, int in_x2, int in_y2, int in_circleX, int in_circleY, int in_radius)
Find intersect points between a line and a circle.

static int Math_FixedPoint_LineRectangleIntersect (int in_x1, int in_y1, int in_x2, int in_y2, int in_rectX, int in_rectY, int in_rectW, int in_rectH)
Find intersect points between a line and a rectangle.

static int Math_FixedPoint_Multiply (int multiplicand, int multiplier)
Multiplication for Fixed Point.

static int Math_FixedPoint_Norm (int x, int y)
Normal of a vector in Fixed Point.

static int Math_FixedPoint_NormPow (int x, int y)
Compute pow of the normal of a vector in Fixed Point.

static int Math_FixedPoint_PointLineDistance (int in_lineX1, int in_lineY1, int in_lineX2, int in_lineY2, int in_pointX, int in_pointY)
Compute the distance from a point to a segment/line.

static int Math_FixedPoint_RadiansToAngleFixedPoint (int angle)
Convert an angle from radians to fixed point angle.

static int Math_FixedPoint_RadiansToDegrees (int angle)
Convert an angle from radians to degrees (both fixed point and not, since this is just a ratio).

static int Math_FixedPoint_Round (int value)
Rounds value to closest whole number in Fixed Point.

static int Math_FixedPoint_Sqr (long value)
Square Root for Long Fixed Point values.

static int Math_FixedPoint_Sqr (int value)
Square Root for Integer Fixed Point values.

static int Math_FixedPoint_Square (int value)
Square for Fixed Point.

static int Math_FixedPoint_Subtract (int minuend, int subtrahend)
Subtraction for Fixed Point.

static int Math_FixedPointAdjust (int a)
Adjust a fixed point in base 8 to this fixed point base.

static int Math_FixedPointAngleToDegree (int a)
Convert a fixed point angle into a degree angle.

static int Math_FixedPointToInt (int a)
Convert a fixed point value to an int value.

static void Math_Init (String packName, int cos, int sqrt) throws Exception
Math initialisation, by reading specified table from a package.

static int Math_IntToFixedPoint (int a)
Convert a fixed point value to an int value.
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math_Log2 (int a)</td>
<td>return log base 2</td>
</tr>
<tr>
<td>Math_Max (long a, long b)</td>
<td>static final long</td>
</tr>
<tr>
<td>Math_Max (int a, int b)</td>
<td>static final int</td>
</tr>
<tr>
<td>Math_Min (long a, long b)</td>
<td>static final long</td>
</tr>
<tr>
<td>Math_Min (int a, int b)</td>
<td>static final int</td>
</tr>
<tr>
<td>Math_Norm (int x, int y, int iter)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_NormPow (int x1, int y1)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_QuickSort (int array[])</td>
<td>static void</td>
</tr>
<tr>
<td>Math_QuickSortIndices (int data[], int nbItemPerValue, int itemNb)</td>
<td>static int[]</td>
</tr>
<tr>
<td>Math_QuickSortIndices (int data[])</td>
<td>static int[]</td>
</tr>
<tr>
<td>Math_Rand ()</td>
<td>static final int</td>
</tr>
<tr>
<td>Math_Rand (int a, int b)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_RandSetSeed (long seed)</td>
<td>static final void</td>
</tr>
<tr>
<td>Math_RectIntersect (int Ax0, int Ay0, int Ax1, int Ay1, int Bx0, int By0, int Bx1, int By1)</td>
<td>static boolean</td>
</tr>
<tr>
<td>Math_SameSign (int a, int b)</td>
<td>static boolean</td>
</tr>
<tr>
<td>Math_SegmentIntersect (int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_Sin (int a)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_Sqrt (long val)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_Sqrt (int x)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_Sqrt_FixedPoint (int val, int precisionLoop)</td>
<td>static int</td>
</tr>
<tr>
<td>Math_Tan (int angle)</td>
<td>static int</td>
</tr>
</tbody>
</table>
Tangent.
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td>Math_Angle180 = Math_DegreeToFixedPointAngle(180) 180 in fixed point</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_Angle270 = Math_DegreeToFixedPointAngle(270) 270 in fixed point</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_Angle360 = Math_DegreeToFixedPointAngle(360) 360 in fixed point</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_Angle90 = Math_DegreeToFixedPointAngle(90) 90 in fixed point</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_AngleMUL = 1 &lt;&lt; GLLibConfig.math_angleFixedPointBase angle fixed point multiplier eg one degree in fixed point base (not equal to Math_DegreeToFixedPointAngle(1))</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_FixedPoint_E = 1459366444 &gt;&gt; (29 - GLLibConfig.math_fixedPointBase)</td>
</tr>
<tr>
<td>static final int</td>
<td>Math_FixedPoint_PI = 1686629713 &gt;&gt; (29 - GLLibConfig.math_fixedPointBase)</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_INTERSECT_NO_INTERSECT = 0</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_INTERSECT_ONE_POINT = (1 &lt;&lt; 1)</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_INTERSECT_TWO_POINTS = (1 &lt;&lt; 2)</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_SEGMENTINTERSECT_COLLINEAR = -1 return value of Math_segmentIntersect if there is no or full intersection because both segment are colinear</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_SEGMENTINTERSECT_DO_INTERSECT = 1 return value of Math_segmentIntersect if there is an intersection</td>
</tr>
<tr>
<td>static final int</td>
<td>MATH_SEGMENTINTERSECT_DONT_INTERSECT = 0 return value of Math_segmentIntersect if there is no intersection</td>
</tr>
<tr>
<td>static int</td>
<td>s_math_bezierX x coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_math_bezierY y coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_math_bezierZ z coordinate of bezier interpolated value through Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_distPointLineX x coordinate of orthogonal projection of point on a segment using Math_distPointLine</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_distPointLineY y coordinate of orthogonal projection of point on a segment using Math_distPointLine</td>
</tr>
</tbody>
</table>
static final int `s_math_F_05 = (s_math_F_1>>1)`
*Fixed point value for number 0.5.*

static final int `s_math_F_1 = 1 << GLLibConfig.math_fixedPointBase`
*Fixed point value for number 1.*

static int `s_Math_intersectPoints[][] = new int[2][2]`

static int `s_Math_intersectX`  
*x coordinate of intersection point of 2 segment using Math_segmentIntersect*

static int `s_Math_intersectY`  
*y coordinate of intersection point of 2 segment using Math_segmentIntersect*

static java.util.Random `s_math_random`
*Random number generator.*
Detailed Description

mathematic function
Function Documentation

static String ConvertFixedPointToString ( int value,  
         int precision  
) [static, package, inherited]

Converts a **GLLib** Fixed Point number to a string in base 10.

**Parameters:**
- *value*  Fixed Point value to be converted.
- *precision*  Number or of decimals to print.

**Returns:**
String representaion of Fixed Point number in base 10

static final long Math_Abs ( long a ) [static, package, inherited]

static final int Math_Abs ( int a ) [static, package, inherited]

static int Math_Atan ( int dx,  
                 int dy  
) [static, package, inherited]

Arctangent.

**Note:**
this function will create a cache to speed up calculation if  
**GLLibConfig.math_AtanUseCacheTable** is set to true, but will also consume up to ((1<<math_fixedPointBase)+1)*4 byte

**Parameters:**
- *dx,dy*  Slope of the angle.

**Returns:**
Arctangent of the slope.
Arctangent, very slow but accurate method, it find angle by dichotomy.

Parameters:
- \( dx, dy \) Slope of the angle.

Returns:
- Arctangent of the slope.

Note:
- This is not the real atan func as found in common device, it will return an angle in \([0, 1]\) and not in \([0, \pi/2]\).

Three control point 2D Bezier interpolation.

Parameters:
- \( x1, y1 \) First control point.
- \( x2, y2 \) Second control point.
- \( x3, y3 \) Third control point.
- \( interp \) Interpolation value (ranges from 0 to \( s\_math\_F\_1 \)).

Returns:
static void Math_Bezier3D ( int \texttt{x1},
   int \texttt{y1},
   int \texttt{z1},
   int \texttt{x2},
   int \texttt{y2},
   int \texttt{z2},
   int \texttt{x3},
   int \texttt{y3},
   int \texttt{z3},
   int \texttt{interp} )
   \texttt{[static, package, inherited]}

Three control point 3D Bezier interpolation.

**Parameters:**
\begin{itemize}
   \item \texttt{x1,y1,z1} First control point.
   \item \texttt{x2,y2,z2} Second control point.
   \item \texttt{x3,y3,z3} Third control point.
   \item \texttt{interp} Interpolation value (ranges from 0 to \texttt{s_math_F_1}).
\end{itemize}

**Returns:**
Result returned in \texttt{s_mathBezierX}, \texttt{s_mathBezierY}, \texttt{s_mathBezierZ}.

static int Math_Cos ( int \texttt{angle} ) \texttt{[static, package, inherited]}

Cosinus.

**Parameters:**
\begin{itemize}
   \item \texttt{angle} - angle in fixed point 0=0 (1<<\texttt{math_angleFixedPointBase})=360.
\end{itemize}

**Returns:**
Cosinus value of the angle in fixed Point

static int Math_DegreeToFixedPointAngle ( int \texttt{a} ) \texttt{[static, package, inherited]}
Convert a degree angle into a fixed point angle.

**Parameters:**
an angle in degrees.

**Returns:**
angle in fixed point.

```java
static int Math_Det ( int x1,
    int y1,
    int x2,
    int y2
)    [static, package, inherited]
```

Compute the determinant of 2 2D vectors.

**Parameters:**
$x1,y1$ X & Y value of the first vector.
$x2,y2$ X & Y value of the second vector.

**Returns:**
The determinant of the 2 vectors.

```java
static int Math_DistPointLine ( int x0,
    int y0,
    int x1,
    int y1,
    int x2,
    int y2
)    throws Exception [static, package, inherited]
```

Compute the distance from a point to a segment/line.

Calculate coordinate of orthogonal projection of point on segment as well.

**Parameters:**
$(x0,y0),(x1,y1)$ Coordinate of segment.
Coordinate of point.

**Returns:**
Distance from point to segment (orthogonal projection) in fixed point.

**Note:**
Coordinates of projected point are saved in
(s_Math_distPointLineX, s_Math_distPointLineY).

```java
static int Math_Div10 ( int a ) [static, package, inherited]
```

Divide by 10 a number, with result in fixed point.

Faster than classical division.

**Parameters:**
a  Number to divide by 10.

**Returns:**
Fixed point value of the number divided by 10.

```java
static int Math_DotProduct ( int x1,
       int y1,
       int x2,
       int y2 ) [static, package, inherited]
```

Compute the dot product of 2 2Dvector.

**Parameters:**
x1,y1  X & Y value of the first vector.
x2,y2  X & Y value of the second vector.

**Returns:**
The dot product of the 2 vectors.
static int Math_FixedPoint_Add ( int summand1, int summand2 ) [static, package, inherited]

Addition for Fixed Point.

**Parameters:**
- *summand1* Number to be added in Fixed Point.
- *summand2* Number to be added in Fixed Point.

**Returns:**
Sum of *summand1* and *summand2* in Fixed Point.

**Note:**
An assert will indicate if the result overflows Integer.MAX_VALUE.

static int Math_FixedPoint_DegreesToAngleFixedPoint ( int angle ) [static, package, inherited]

Convert an angle from degrees to fixed point angle.

**Parameters:**
- *angle* in degrees to be converted.

**Returns:**
angle in Fixed Point.

static int Math_FixedPoint_DegreesToRadians ( int angle ) [static, package, inherited]

Convert an angle from degrees to radians (both fixed point and not, since this is just a ratio).

**Parameters:**
- *angle* in degrees to be converted into radians.

**Returns:**
angle in radians. (same precision as input param)
Compute the determinant of 2 2D vector in Fixed Point.

**Parameters:**
- $x1, y1$ X & Y value of the first vector in Fixed Point.
- $x2, y2$ X & Y value of the second vector in Fixed Point.

**Returns:**
The determinant of the 2 vectors in Fixed Point.

Division for Fixed Point.

**Parameters:**
- `dividend` Number to be divided in Fixed Point.
- `divisor` Number to be divided by in Fixed Point.

**Returns:**
Quotient of dividend and divisor in Fixed Point.

**Note:**
An assert will indicate a division by zero.

Compute the dot product of 2 2D vector in Fixed Point.
Parameters:

\(x_1, y_1\) X & Y value of the first vector in Fixed Point.

\(x_2, y_2\) X & Y value of the second vector in Fixed Point.

Returns:
The dot product of the 2 vectors in Fixed Point.

---

```c
static int Math_FixedPoint_LineCircleIntersect ( int in_x1,
    int in_y1,
    int in_x2,
    int in_y2,
    int in_circleX,
    int in_circleY,
    int in_radius )
```

Find intersect points between a line and a circle.

Parameters:

\((\text{in}_x_1, \text{in}_y_1)\) first point on the line.

\((\text{in}_x_2, \text{in}_y_2)\) second point on the line.

\((\text{in}_\text{circle}_X, \text{in}_\text{circle}_Y)\) center coordinates of circle.

\((\text{in}_\text{radius})\) radius of the circle.

Returns:

- MATH_INTERSECT_NO_INTERSECT if no intersect between line and circle.
- MATH_INTERSECT_ONE_POINT if only one intersect point between line and circle (Tangeant).
- MATH_INTERSECT_TWO_POINTS if two intersect points between line and circle.

Note:
Coordinates of intersection points are saved in the array s_Math_intersectPoints[][] First index indicates point: 0 = first point and 1 = second point Second index indicates x or y coordinate: 0 = x coordinate and 1 = y coordinate

```c
static int Math_FixedPoint_LineRectangleIntersect ( int in_x1,
```
Find intersect points between a line and a rectangle.

**Parameters:**

- 
  
  (in\_x1,in\_y1) \hspace{1cm} \text{first point on the line.}
  
  (in\_x2,in\_y2) \hspace{1cm} \text{second point on the line.}
  
  (in\_rectX,in\_rectY) \hspace{1cm} \text{coordinates of top-left corner of rectangle.}
  
  (in\_rectW,in\_rectH) \hspace{1cm} \text{Width and Height of rectangle.}

**Returns:**

- MATH\_INTERSECT\_NO\_INTERSECT if no intersect between line and rectangle.
- MATH\_INTERSECT\_ONE\_POINT if only one intersect point between line and rectangle.
- MATH\_INTERSECT\_TWO\_POINTS if two intersect points between line and circle.

**Note:**

Coordinates of intersection points are saved in the array s\_Math\_intersectPoints[][] First index indicates point: 0 = first point and 1 = second point Second index indicates x or y coordinate: 0 = x coordinate and 1 = y coordinate

```c
static int Math\_FixedPoint\_Multiply ( int multiplicand,
                                          int multiplier
                                      )
                                      [static, package, inherited]
```

**Multiplication for Fixed Point.**

**Parameters:**

- multiplicand \hspace{1cm} \text{Number to be multiplied in Fixed Point.}
- multiplier \hspace{1cm} \text{Number to be multiplied by in Fixed Point.}
Returns:
Product of multiplicand and multiplier in Fixed Point.

Note:
An assert will indicate if the result overflows Integer.MAX_VALUE.

```java
static int Math_FixedPoint_Norm ( int x,
    int y
  )  [static, package, inherited]
```

Normal of a vector in Fixed Point.

Parameters:
  x,y X & Y value of the vector in Fixed Point.

Returns:
  norm of vector (x,y).

```java
static int Math_FixedPoint_NormPow ( int x,
    int y
  )  [static, package, inherited]
```

Compute pow of the normal of a vector in Fixed Point.

Parameters:
  x,y X & Y value of the vector in Fixed Point.

Returns:
  The pow of the normal of the vector in Fixed Point.

Note:
An assert will indicate if the result overflows Integer.MAX_VALUE.

```java
static int Math_FixedPoint_PointLineDistance ( int in_lineX1,
    int in_lineY1,
    int in_lineX2,
    int in_lineY2,
  )  [static, package, inherited]
```
Compute the distance from a point to a segment/line.

Calculate coordinate of orthogonal projection of point on segment as well.

**Parameters:**

- `(in_lineX1,in_lineY1)(in_lineX2,in_lineY2)` Coordinate of segment.
- `(in_pointX,in_pointY)` Coordinate of point.

**Returns:**

Distance from point to segment (orthogonal projection) in fixed point.

**Note:**

Coordinates of intersection points in fixed Point are saved in `(s_Math_distPointLineX, s_Math_distPointLineY)`.

---

Convert an angle from radians to fixed point angle.

**Parameters:**

- `angle` in radians to be converted.

**Returns:**

`angle` in Fixed Point.

---

Convert an angle from radians to degrees (both fixed point and not, since this is just a ratio).

**Parameters:**

- `angle` in radians to be converted into degrees.
Returns:
angle in degress. (same precision as input param)

static int Math_FixedPoint_Round ( int value ) [static, package, inherited]

Rounds value to closest whole number in Fixed Point.

Parameters:
value Value to be rounded in Fixed Point.

Returns:
Rounded value in Fixed Point.

static int Math_FixedPoint_Sqrt ( long value ) [static, package, inherited]

Square Root for Long Fixed Point values.

Parameters:
value Number to get square root of.

Returns:
Square root of val in Fixed Point.

Note:
Iterative process, precise but can be slow, use with caution.

static int Math_FixedPoint_Sqrt ( int value ) [static, package, inherited]

Square Root for Integer Fixed Point values.

Parameters:
value Number to get square root of.

Returns:
Square root of val in Fixed Point.
**Note:**
Iterative process, precise but can be slow, use with caution.

```java
static int Math_FixedPoint_Square ( int value ) [static, package, inherited]
```

Square for Fixed Point.

**Parameters:**
- `value` Number to be squared in Fixed Point.

**Returns:**
Squared value in Fixed Point.

```java
static int Math_FixedPoint_Subtract ( int minuend, int subtrahend ) [static, package, inherited]
```

Subtraction for Fixed Point.

**Parameters:**
- `minuend` Number to be subtracted from in Fixed Point.
- `subtrahend` Number to be subtracted in Fixed Point.

**Returns:**
Difference of minuend and subtrahend in Fixed Point.

**Note:**
An assert will indicate if the result overflows Integer.MAX_VALUE.

```java
static int Math_FixedPointAdjust ( int a ) [static, package, inherited]
```

Adjust a fixed point in base 8 to this fixed point base.

**Parameters:**
- `a` Value to adjust.
**Returns:**
The value adjusted.

```java
static int Math_FixedPointAngleToDegree ( int a ) [static, package, inherited]
```

Convert a fixed point angle into a degree angle.

**Parameters:**
- `a` angle in fixed point.

**Returns:**
angle in degrees.

```java
static int Math_FixedPointToInt ( int a ) [static, package, inherited]
```

Convert a fixed point value to an int value.

**Parameters:**
- `a` Fixed point value to convert to int.

**Returns:**
Int value of `a`.

```java
static void Math_Init ( String packName,
                        int cos,
                        int sqrt ) throws Exception [static, package, inherited]
```

Math initialisation, by reading specified table from a package.

**Parameters:**
- `packName` Name of the pack which contains math table.
- `cos` Index of cosine table in the pack. Use -1 if no table but all call to Math_Cos & Math_Sin will fail.
- `sqrt` Index of sqrt table in the pack. Use -1 if no table but all call to Math_sqrt will fail.
Returns:
    True if everything was loaded as requested.

static int Math_IntToFixedPoint ( int a ) [static, package, inherited]

Convert a fixed point value to an int value.

Parameters:
    a  Value to convert to fixed point.

Returns:
    Fixed point value of a.

static int Math_Log2 ( int a ) [static, package, inherited]

return log base 2

Parameters:
    a  value to calculate log base 2

Returns:
    log base 2 of value

static final long Math_Max ( long a,
    long b
) [static, package, inherited]

static final int Math_Max ( int a,
    int b
) [static, package, inherited]

static final long Math_Min ( long a,
    long b
) [static, package, inherited]
static final int Math_Min ( int a,
                   int b
               )   [static, package, inherited]

static int Math_Norm ( int x,
                  int y,
                  int iter
              )   [static, package, inherited]

Norm following newton law approximation.

Sqrt(n) is provided by iteration of xk = (xk + n / xk) / 2. The more iteration (k++), the better is the approximation.

Parameters:
         x,y  X and Y coordinate of vector to get the norm from.
         iter Number of iteration to perform, more iteration gives a better result however more iteration are more cpu intensive.

Returns:
    norm of vector (x,y).

static int Math_NormPow ( int x1,
                  int y1
              )   [static, package, inherited]

Compute pow of the normal of a vector.

Parameters:
      x1,y1  X & Y value of the vector.

Returns:
    The pow of the normal of the vector.

static void Math_QuickSort ( int array[] ) [static, package, inherited]
Quicksort an array of integer.

**Parameters:**
- array Array of data to sort.

```java
static int[] Math_QuickSortIndices ( int data[], int nbItemPerValue, int itemNb )
```

Get an array of indice corresponding to the sorting of an array of data.

Example: data={12, 3, 8} return {1, 2, 0} so that data[1] <= data[2] <= data[0].

**Parameters:**
- data Array of data to scale.
- nbItemPerValue If data is an array of values, how many item are needed per value.
- itemNb If data is an array of values, which item shoudl be considered for sorting.

**Returns:**
- Array of indices, so that data[indices] are sorted.

```java
static int[] Math_QuickSortIndices ( int data[] )
```

Get an array of indice corresponding to the sorting of an array of data.

Example: data={12, 3, 8} return {1, 2, 0} so that data[1] <= data[2] <= data[0].

**Parameters:**
- data Array of data to scale.

**Returns:**
- Array of indices, so that data[indices] are sorted.
## static void Math_Quit () [static, package, inherited]

Free all math arrays.

## static final int Math_Rand () [static, package, inherited]

Create a random number.

**Returns:**
- A random int.

## static int Math_Rand ( int \( a \), int \( b \)) [static, package, inherited]

Create a random int inside the interval \([a, b]\) [a, b[.

**Parameters:**
- \( a, b \) Interval for the random number to generate.

**Returns:**
- A number between \([a, b]\) [a, b[.

**Note:**
- if \( a=b \) then return value is \( a \) (or \( b \))

## static final void Math_RandSetSeed ( long \( seed \)) [static, package, inherited]

set math random seed

**Parameters:**
- \( seed \) seed to use for random number generator.

## static boolean Math_RectIntersect ( int \( Ax0, \)
  int \( Ay0, \)
Tell if 2 axis aligned rectangle intersect.

**Parameters:**

\[(Ax0,Ay0)(Ax1,Ay1)\] top left and bottom right coordinate of first rectangle  
\[(Bx0,By0)(Bx1,By1)\] top left and bottom right coordinate of second rectangle

**Returns:**

true if intersect, false if not

---

Test 2 numbers to see if they are both of the same sign.

**Parameters:**

\[a,b\] Numbers to compare.

**Returns:**

True if both numbers are positive.  
True if both numbers are negative.  
False Otherwise.

---

Test 2 numbers to see if they are both of the same sign.
Tell if 2 segment intersect themselves.

**Parameters:**

\((x_1, y_1)(x_2, y_2)\) Coordinate of first segment.
\((x_3, y_3)(x_4, y_4)\) Coordinate of second segment.

**Returns:**

- `MATH_SEGMENTINTERSECT_DO_INTERSECT` if intersect.
- `MATH_SEGMENTINTERSECT_DOинтерSECT` if no intersection.
- `MATH_SEGMENTINTERSECT_COLLINEAR` if two segment are colinear (full or no intersection).

**Note:**
Coordinates of intersection point are saved in \((s\_Math\_intersectX, s\_Math\_intersectY)\).

**Warning:**
This function can generate some Overflows if large value are used. Be careful when using this code with FixedPoint values, the boundaries can change with the value of `GLLibConfig.math\_fixedPointBase`.

```cpp
static int Math_Sin ( int a ) [static, package, inherited]
Sinus.

**Parameters:**

\(a\) Angle in fixed point 0=0 (1<<math\_angleFixedPointBase)=360.

**Returns:**

Sinus value of the angle.

```
Square Root slow.

**Parameters:**

val  Number to get square root of.

**Returns:**

Square root of val.

**Note:**
long version is way slower than int function

```java
static int Math_Sqrt (int x) [static, package, inherited]
```

Square Root.

**Parameters:**

x  Number to get square root of.

**Returns:**

Square root of x.

**Note:**
Integer Square Root function. Contributors include Arne Steinarson for the basic approximation idea, Dann Corbit and Mathew Hendry for the first cut at the algorithm, Lawrence Kirby for the rearrangement, improvements and range optimization, Paul Hsieh for the round-then-adjust idea, and Tim Tyler, for the Java port.

```java
static int Math_Sqrt_FixedPoint (int val,
                                 int precisionLoop
                              ) [static, package, inherited]
```

Square Root for Fixed Point.

**Parameters:**

val  Number to get square root of.
**precisionLoop**

Number of time to go through the algo to get good precision in the result. Good value are [10-30] 15 is usually good.

**Returns:**
Square root of val in Fixed Point.

**Note:**
Iterative process, can be slow, use with caution.

```java
static int Math_Tan ( int angle ) [static, package, inherited]
```

Tangent.

**Parameters:**

*angle* - Angle in fixed point 0=0 256=360

**Returns:**
Tangent value of the angle.
Variable Documentation

final int Math_Angle180 = Math_DegreeToFixedPointAngle(180) [static, package, inherited]
180 in fixed point

final int Math_Angle270 = Math_DegreeToFixedPointAngle(270) [static, package, inherited]
270 in fixed point

final int Math_Angle360 = Math_DegreeToFixedPointAngle(360) [static, package, inherited]
360 in fixed point

final int Math_Angle90 = Math_DegreeToFixedPointAngle(90) [static, package, inherited]
90 in fixed point

final int Math_AngleMUL = 1 << GLLibConfig.math_angleFixedPointBase [static, package, inherited]
angle fixed point multiplier eg one degree in fixed point base.(not equal to Math_DegreeToFixedPointAngle(1))

final int Math_FixedPoint_E = 1459366444 >> (29 - GLLibConfig.math_fixedPointBase) [static, package, inherited]

final int Math_FixedPoint_PI = 1686629713 >> (29 - GLLibConfig.math_fixedPointBase) [static, package, inherited]

final int MATH_INTERSECT_NO_INTERSECT = 0 [static, package, inherited]
final int MATH_INTERSECT_ONE_POINT = (1 << 1) [static, package, inherited]

final int MATH_INTERSECT_TWO_POINTS = (1 << 2) [static, package, inherited]

final int MATH_SEGMENTINTERSECT_COLLINEAR = -1 [static, package, inherited]

return value of Math_segmentIntersect if there is no or full intersection because both segment are colinear

final int MATH_SEGMENTINTERSECT_DO_INTERSECT = 1 [static, package, inherited]

return value of Math_segmentIntersect if there is an intersection

final int MATH_SEGMENTINTERSECT_DONT_INTERSECT = 0 [static, package, inherited]

return value of Math_segmentIntersect if there is no intersection

int s_math_bezierX [static, package, inherited]

x coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D

int s_math_bezierY [static, package, inherited]

y coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D

int s_math_bezierZ [static, package, inherited]

z coordinate of bezier interpolated value through Math_Bezier3D
int s_Math_distPointLineX [static, package, inherited]

x coordinate of orthogonal projection of point on a segment using Math_distPointLine

int s_Math_distPointLineY [static, package, inherited]

y coordinate of orthogonal projection of point on a segment using Math_distPointLine

final int s_math_F_05 = (s_math_F_1>>1) [static, package, inherited]

Fixed point value for number 0.5.

final int s_math_F_1 = 1 << GLLibConfig.math_fixedPointBase [static, package, inherited]

Fixed point value for number 1.

int s_Math_intersectPoints[][] = new int[2][2] [static, package, inherited]

int s_Math_intersectX [static, package, inherited]

x coordinate of intersection point of 2 segment using Math_segmentIntersect

int s_Math_intersectY [static, package, inherited]

y coordinate of intersection point of 2 segment using Math_segmentIntersect
Random number generator.
GLLib : File/Package System

[GLLib]

handle data file access (read only) More...
## Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static String GetMIME (int idx)</code></td>
<td>Get the MIME type as a String for a MIME index.</td>
</tr>
<tr>
<td><code>static int GetNBData ()</code></td>
<td>Get the number of data in current data file.</td>
</tr>
<tr>
<td><code>InputString GetResourceAsStream (String s)</code></td>
<td>Get the number of data in current data file.</td>
</tr>
<tr>
<td><code>static void Pack_Close ()</code></td>
<td>Close a pack file.</td>
</tr>
<tr>
<td><code>static void Pack_LoadMIME (String filename)</code></td>
<td>Load the MIME type from a MIME Pack.</td>
</tr>
<tr>
<td><code>static void Pack_Open (String filename)</code></td>
<td>Open a pack given its path and filename.</td>
</tr>
<tr>
<td><code>static int Pack_PositionAtData (int idx)</code></td>
<td>Position the pack pointer to the begining of data idx.</td>
</tr>
<tr>
<td><code>static int Pack_Read ()</code></td>
<td>Read a single byte from the current stream.</td>
</tr>
<tr>
<td><code>static int Pack_Read16 ()</code></td>
<td>Read one unsigned short from the current stream.</td>
</tr>
<tr>
<td><code>static int Pack_Read32 ()</code></td>
<td>Read one int from the current stream.</td>
</tr>
<tr>
<td><code>static Object Pack_ReadArray (int idx)</code></td>
<td>Read an array or multiarray from the stream.</td>
</tr>
<tr>
<td><code>static byte[] Pack_ReadData (int idx)</code></td>
<td>Read and return the data at idx.</td>
</tr>
<tr>
<td><code>static int Pack_ReadFully (byte[] array, int offset, int length)</code></td>
<td>Read into a byte array.</td>
</tr>
<tr>
<td><code>static void Pack_ReleaseBinaryCache (String filename)</code></td>
<td>Release cache data of a pack and subpack.</td>
</tr>
<tr>
<td><code>static void Pack_Seek (int addr)</code></td>
<td>Set the current offset to addr byte from the beginning.</td>
</tr>
<tr>
<td><code>static void Pack_Skip (int nb)</code></td>
<td>Skip ahead in the current stream.</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static byte[][]</td>
<td><strong>MIME_type</strong></td>
<td>MIME type array. Stored as UTF-8.</td>
</tr>
<tr>
<td>static int</td>
<td><strong>s_pack_lastDataReadMimeType</strong></td>
<td>Mime type of last data that have been read through Pack_Read.</td>
</tr>
</tbody>
</table>
Detailed Description

handle data file access (read only)

Packages are composed of at least one data file (subpack).

The first data file (first subpack) contain extra information about all the data stored within the pack.

Extra information for 1st subpack, at the very beginning of the file:

- 2 byte : Number of data within the pack. How many data are store within this pack (including all subpack)).
- 2 byte : Number of subpacks. How many subpacks compose this pack.
- 2 byte * number of subpack (fat table): List the starting data number of each subpack.
  Example with 3 sub pack we could have : 0, 8, 15 :
  - Subpack0 contain data(0) to data(7).
  - Subpack1 contain data(8), data(14).
  - Subpack2 contain data(15) to data(nb of data).

Then each subpack (including the first one) contain:

- 4 byte * number of data in this subpack + 1 : offset table. This offset is the offset of each data from the begining of this file to the data. The +1 is for an extra value which represent the end of the file, this is used to calculate the size of each data within the subpack (size of data) = (offset of next data) - (offset of the data).

Finally, for each data inside the subpack:

- 1 byte : mime type of the data except if file is a dummy file (eg size = 0)
- x byte : the data

There is also one "special" pack, which contain the MIME type if some
were defined:

- 1 byte: Mime type count.
- For each mime type:
  - 1 byte: length of mime type byte array.
  - Length of mime type byte: byte array encoded as UTF-8 representing the mime type.
## Function Documentation

### static String GetMIME ( int idx ) [static, package, inherited]

Get the MIME type as a String for a MIME index.

**Parameters:**
- `idx`: Index to get MIME type.

**Returns:**
- String containing the requested MIME type or empty string.

### static int GetNBData ( ) [static, package, inherited]

Get the number of data in current data file.

**Returns:**
- the number of data

### InputStream GetResourceAsStream ( String s ) [inherited]

### static void Pack_Close ( ) [static, package, inherited]

Close a pack file.

Nothing will be done if there are no pack currently open.

**Note:**
- This is the only way to free the memory consumed by a pack loaded in memory. `GLib.Gc()` will be called by this function.

### static void Pack_LoadMIME ( String filename ) [static, package, inherited]

Load the MIME type from a MIME Pack.
**Parameters:**

*filename*  Filename of the pack containing the MIME type.

---

```java
static void Pack_Open ( String filename ) [static, package, inherited]
```

Open a pack given its path and filename.

Example: Pack_Open("/0");

**Parameters:**

*filename*  Filename of the package to open.

**Note:**

This load the whole pack in memory watch for phone constraint.

---

```java
static int Pack_PositionAtData ( int idx ) [static, package, inherited]
```

Position the pack pointer to the begining of data idx.

If the data requested is not in this subpack, the correct subpack will be opened. This can be usefull if you want to read/stream_parse the pack-stream by yourself.

**Returns:**

Length of the requested data.

**Note:**

s_pack_lastDataReadMimeType is now valid for the requested data.

---

```java
static int Pack_Read ( ) [static, package, inherited]
```

Read a single byte from the current stream.

**Returns:**

one byte as a int, or -1 if EOF.
**Exceptions:**

*Exception* if error occurred.

---

**static int Pack_Read16 ( ) [static, package, inherited]**

Read one unsigned short from the current stream.

LittleEndian Format.

**Returns:**

Return unsigned short value as an int.

**Exceptions:**

*Exception* if error occurred.

---

**static int Pack_Read32 ( ) [static, package, inherited]**

Read one int from the current stream.

LittleEndian Format.

**Returns:**

Return int value.

**Exceptions:**

*Exception* if error occurred.

---

**static Object Pack_ReadArray ( int idx ) [static, package, inherited]**

Read an array or multiarray from the stream.

ID is encoded as:

- bit 0-1 = type of data array (ARRAY_BYTE, ARRAY_SHORT, ARRAY_INT).
- bit 2 = 1 if multidimension array, 0 otherwise.
- bit 3 = encoding for nb of component in array (0=byte, 1 = short).
- bit 4-5 = if multidimensional array . dimension of array else real padding of data inside array.
- bit 6-7 = unused.

**Parameters:**

`idx` Index of the array/multiarray to read.

**Returns:**

Array or multiarray as an Object.

**Exceptions:**

`Exception` if error occured.

```java
static byte [] Pack_ReadData ( int idx ) [static, package, inherited]
```

Read and return the data at idx.

If needed, the system, will seek to the data, or even open another subpack.

**Parameters:**

`idx` Index of the data to read in this package.

**Returns:**

A byte array containing the data requested.

```java
static int Pack_ReadFully ( byte[] array, int offset, int length ) [static, package, inherited]
```

Read into a byte array.

**Parameters:**

`array` Array to store data read, array must be non null.
offset  Offset to store byte in array.
length  Number of byte to read.

Returns:
Number of byte read, must be length or exception.

**static void Pack_ReleaseBinaryCache ( String filename )** [static, package, inherited]

Release cache data of a pack and subpack.

**Parameters:**

  filename  Filename of the package to release.

**static void Pack_Seek ( int addr )** [static, package, inherited]

Set the current offset to addr byte from the beginning.

**Parameters:**

  addr  Offset to seek to from the begining of the current stream.

**static void Pack_Skip ( int nb )** [static, package, inherited]

Skip ahead in the current stream.

Move forward the current offset in the stream.

**Parameters:**

  nb  Number of byte to skip in data file.

**Exceptions:**

  Exception  if error occurred.
**Variable Documentation**

<table>
<thead>
<tr>
<th>Name</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte [] MIME_type</td>
<td>MIME type array. Stored as UTF-8.</td>
</tr>
<tr>
<td>int s_pack_lastDataReadMimeType</td>
<td>Mime type of last data that have been read through Pack_Read.</td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:30 2008 for GLLib by doxygen 1.5.2*
GLLib : MIDP wrapper

[GLLib]

wrapper for some usual midp function, use this instead of the midp ones

More...
# Functions

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClipRect</td>
<td>Intersects the current clip with the specified rectangle.</td>
</tr>
<tr>
<td>CopyArea</td>
<td>Copies the contents of a rectangular area ((x_{\text{src}}, y_{\text{src}}, \text{width}, \text{height})) to a destination area, whose anchor point identified by anchor is located at ((x_{\text{dest}}, y_{\text{dest}})).</td>
</tr>
<tr>
<td>DrawArc</td>
<td>Draws the outline of a circular or elliptical arc covering the specified rectangle, using the current color and stroke style.</td>
</tr>
<tr>
<td>DrawChar</td>
<td>Draws the specified character using the current font and color.</td>
</tr>
<tr>
<td>DrawChars</td>
<td>Draws the specified characters using the current font and color.</td>
</tr>
<tr>
<td>DrawImage</td>
<td>Draws the specified image by using the anchor point.</td>
</tr>
<tr>
<td>DrawLine</td>
<td>Draws a line between the coordinates ((x_{1}, y_{1})) and ((x_{2}, y_{2})) using the current color and stroke style.</td>
</tr>
<tr>
<td>drawPartialRGB</td>
<td>Same as drawPartialRGB but it's useful when you draw in a back buffer that is bigger than the screen.</td>
</tr>
<tr>
<td>drawPartialRGB</td>
<td>Same as drawRGB but it draws safely (without drawing outside the screen).</td>
</tr>
<tr>
<td>DrawRect</td>
<td>Draws the outline of the specified rectangle using the current color and stroke style.</td>
</tr>
<tr>
<td>DrawRegion</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Parameters</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>static final void x_src, int y_src, int width, int height, int transform, int x_dest, int y_dest, int anchor</td>
<td>x_src, y_src, width, height, transform, x_dest, y_dest, anchor</td>
</tr>
<tr>
<td>static final void DrawRGB (int[] rgbData, int offset, int scanlength, int x, int y, int width, int height, boolean processAlpha)</td>
<td>rgbData, offset, scanlength, x, y, width, height, processAlpha</td>
</tr>
<tr>
<td>static final void DrawRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)</td>
<td>x, y, width, height, arcWidth, arcHeight</td>
</tr>
<tr>
<td>static final void DrawString (String str, int x, int y, int anchor)</td>
<td>str, x, y, anchor</td>
</tr>
<tr>
<td>static final void DrawSubstring (String str, int offset, int len, int x, int y, int anchor)</td>
<td>str, offset, len, x, y, anchor</td>
</tr>
<tr>
<td>static final void FillArc (int x, int y, int width, int height, int startAngle, int arcAngle)</td>
<td>x, y, width, height, startAngle, arcAngle</td>
</tr>
<tr>
<td>static final void FillRect (int x, int y, int width, int height)</td>
<td>x, y, width, height</td>
</tr>
<tr>
<td>static final void FillRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)</td>
<td>x, y, width, height, arcWidth, arcHeight</td>
</tr>
<tr>
<td>static final void FillTriangle (int x1, int y1, int x2, int y2, int x3, int y3)</td>
<td>x1, y1, x2, y2, x3, y3</td>
</tr>
<tr>
<td>static final void Gc ()</td>
<td></td>
</tr>
<tr>
<td>static final int GetBlueComponent ()</td>
<td></td>
</tr>
<tr>
<td>static final int GetClipHeight ()</td>
<td></td>
</tr>
<tr>
<td>static final int GetClipWidth ()</td>
<td></td>
</tr>
<tr>
<td>static final int GetClipX ()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>static final int</strong> GetClipY ()</td>
<td>Gets the Y offset of the current clipping area, relative to the coordinate system origin of this graphics context.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetColor ()</td>
<td>Gets the current color.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetDisplayColor (int color)</td>
<td>Gets the color that will be displayed if the specified color is requested.</td>
</tr>
<tr>
<td><strong>static final javax.microedition.lcdui.Font</strong> GetFont ()</td>
<td>Gets the current font.</td>
</tr>
<tr>
<td><strong>static final long</strong> GetFrameTime ()</td>
<td>Get time for this frame (real time when this frame started)</td>
</tr>
<tr>
<td><strong>static final int</strong> GetGrayScale ()</td>
<td>Gets the current grayscale value of the color being used for rendering operations.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetGreenComponent ()</td>
<td>Gets the green component of the current color.</td>
</tr>
<tr>
<td><strong>static final long</strong> GetRealTime ()</td>
<td>Get real time at the moment the function is called</td>
</tr>
<tr>
<td><strong>static final int</strong> GetRedComponent ()</td>
<td>Gets the red component of the current color.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetScreenHeight ()</td>
<td>Get screen height, (if screen orientation changes, it will return the new screen height automatically</td>
</tr>
<tr>
<td><strong>static final int</strong> GetScreenWidth ()</td>
<td>Get screen width, (if screen orientation changes, it will return the new screen width automatically</td>
</tr>
<tr>
<td><strong>static final int</strong> GetStrokeStyle ()</td>
<td>Gets the stroke style used for drawing operations.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetTranslateX ()</td>
<td>Gets the X coordinate of the translated origin of this graphics context.</td>
</tr>
<tr>
<td><strong>static final int</strong> GetTranslateY ()</td>
<td>Gets the Y coordinate of the translated origin of this graphics context.</td>
</tr>
<tr>
<td><strong>static final boolean</strong> PlatformRequest (String url)</td>
<td>Performs platformRequest.</td>
</tr>
<tr>
<td><strong>static final void</strong> SetClip (int x, int y, int width, int height)</td>
<td>Sets the current clip to the rectangle specified by the given coordinates.</td>
</tr>
</tbody>
</table>
| static final void | **setColor** (int red, int green, int blue)  
|                  | *Sets the current color to the specified RGB values.* |
| static final void | **SetColor** (int RGB)  
|                  | *Sets the current color to the specified RGB values.* |
| static final void | **SetCurrentGraphics** (javax.microedition.lcdui.Image img)  
|                  | *set current graphics context* |
| static final void | **SetCurrentGraphics** (javax.microedition.lcdui.Graphics graphics)  
|                  | *set current graphics context* |
| static final void | **SetFont** (javax.microedition.lcdui.Font font)  
|                  | *Sets the font for all subsequent text rendering operations.* |
| static final void | **SetGrayScale** (int value)  
|                  | *Sets the current grayscale to be used for all subsequent rendering operations.* |
| static final void | **SetStrokeStyle** (int style)  
|                  | *Sets the stroke style used for drawing lines, arcs, rectangles, and rounded rectangles.* |
| static final void | **Translate** (int x, int y)  
|                  | *Translates the origin of the graphics context to the point (x, y) in the current coordinate system.* |
| static void      | **Vibrate** (int duration)  
|                  | *Make the phone vibrate or flash the back light is the phone has this functionnality.* |
**Variables**

<table>
<thead>
<tr>
<th>Static final int</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOTTOM = 32</td>
<td>Constant for positioning the anchor point of text and images below the text or image.</td>
<td></td>
</tr>
<tr>
<td>DOTTED = 1</td>
<td>Constant for the DOTTED stroke style.</td>
<td></td>
</tr>
<tr>
<td>HCENTER = 1</td>
<td>Constant for centering text and images horizontally around the anchor point.</td>
<td></td>
</tr>
<tr>
<td>LEFT = 4</td>
<td>Constant for positioning the anchor point of text and images to the left of the text or image.</td>
<td></td>
</tr>
<tr>
<td>RIGHT = 8</td>
<td>Constant for positioning the anchor point of text and images to the right of the text or image.</td>
<td></td>
</tr>
<tr>
<td>SOLID = 0</td>
<td>Constant for the SOLID stroke style.</td>
<td></td>
</tr>
<tr>
<td>TOP = 16</td>
<td>Constant for positioning the anchor point of text and images above the text or image.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR = 2</td>
<td>causes the specified image region to be reflected about its vertical center.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT180 = 1</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 180 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT270 = 4</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 270 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT90 = 7</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 90 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_NONE = 0</td>
<td>No transform is applied.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT180 = 3</td>
<td>causes the specified image region to be rotated clockwise by 180 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT270 = 6</td>
<td>causes the specified image region to be rotated clockwise by 270 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT90 = 5</td>
<td>causes the specified image region to be rotated clockwise by 90 degrees.</td>
<td></td>
</tr>
<tr>
<td>VCENTER = 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constant for centering text and images vertically around the anchor point.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

wrapper for some usual midp function, use this instead of the midp ones
Function Documentation

static final void ClipRect ( int  x,
    int  y,
    int  width,
    int  height
  )

Intersects the current clip with the specified rectangle.

Parameters:
  x - the x coordinate of the rectangle to intersect the clip with
  y - the y coordinate of the rectangle to intersect the clip with
  width - the width of the rectangle to intersect the clip with
  height - the height of the rectangle to intersect the clip with

See also:
  setClip

static final void CopyArea ( int  x_src,
    int  y_src,
    int  width,
    int  height,
    int  x_dest,
    int  y_dest,
    int  anchor
  )

Copies the contents of a rectangular area (x_src, y_src, width, height) to a destination area, whose anchor point identified by anchor is located at (x_dest, y_dest).

The effect must be that the destination area contains an exact copy of the contents of the source area immediately prior to the invocation of this method. This result must occur even if the source and destination areas overlap.

The points (x_src, y_src) and (x_dest, y_dest) are both specified
relative to the coordinate system of the Graphics object. It is illegal for the source region to extend beyond the bounds of the graphic object. This requires that:
\[\begin{align*}
    x_{\text{src}} + tx & \geq 0 \\
    y_{\text{src}} + ty & \geq 0 \\
    x_{\text{src}} + tx + \text{width} & \leq \text{width of Graphics object's destination} \\
    y_{\text{src}} + ty + \text{height} & \leq \text{height of Graphics object's destination}
\end{align*}\]

where \(tx\) and \(ty\) represent the X and Y coordinates of the translated origin of this graphics object, as returned by \(\text{getTranslateX()}\) and \(\text{getTranslateY()}\), respectively.

However, it is legal for the destination area to extend beyond the bounds of the Graphics object. Pixels outside of the bounds of the Graphics object will not be drawn.

The \text{copyArea} method is allowed on all Graphics objects except those whose destination is the actual display device. This restriction is necessary because allowing a \text{copyArea} method on the display would adversely impact certain techniques for implementing double-buffering.

Like other graphics operations, the \text{copyArea} method uses the Source Over Destination rule for combining pixels. However, since it is defined only for mutable images, which can contain only fully opaque pixels, this is effectively the same as pixel replacement.

**Parameters:**

- \(x_{\text{src}}\) - the x coordinate of upper left corner of source area
- \(y_{\text{src}}\) - the y coordinate of upper left corner of source area
- \(width\) - the width of the source area
- \(height\) - the height of the source area
- \(x_{\text{dest}}\) - the x coordinate of the destination anchor point
- \(y_{\text{dest}}\) - the y coordinate of the destination anchor point
- \(anchor\) - the anchor point for positioning the region within the destination image

```java
static final void DrawArc ( int x,  
    int y,  
    int width,
```
int height,
int startAngle,
int arcAngle
)

[static, package, inherited]

Draws the outline of a circular or elliptical arc covering the specified rectangle, using the current color and stroke style.

The resulting arc begins at startAngle and extends for arcAngle degrees, using the current color. Angles are interpreted such that 0 degrees is at the 3 o'clock position. A positive value indicates a counter-clockwise rotation while a negative value indicates a clockwise rotation.

The center of the arc is the center of the rectangle whose origin is (x, y) and whose size is specified by the width and height arguments.

The resulting arc covers an area width + 1 pixels wide by height + 1 pixels tall. If either width or height is less than zero, nothing is drawn.

The angles are specified relative to the non-square extents of the bounding rectangle such that 45 degrees always falls on the line from the center of the ellipse to the upper right corner of the bounding rectangle. As a result, if the bounding rectangle is noticeably longer in one axis than the other, the angles to the start and end of the arc segment will be skewed farther along the longer axis of the bounds.

Parameters:

- x - the x coordinate of the upper-left corner of the arc to be drawn
- y - the y coordinate of the upper-left corner of the arc to be drawn
- width - the width of the arc to be drawn
- height - the height of the arc to be drawn
- startAngle - the beginning angle
- arcAngle - the angular extent of the arc, relative to the start angle
Draws the specified character using the current font and color.

Parameters:

- `character` - the character to be drawn
- `x` - the x coordinate of the anchor point
- `y` - the y coordinate of the anchor point
- `anchor` - the anchor point for positioning the text; see anchor points

```java
static final void DrawChars ( char[] data, int offset, int length, int x, int y, int anchor )
```

Draws the specified characters using the current font and color.

The offset and length parameters must specify a valid range of characters within the character array data. The offset parameter must be within the range `[0..(data.length)]`, inclusive. The length parameter must be a non-negative integer such that `(offset + length) <= data.length`.

Parameters:

- `data` - the array of characters to be drawn
- `offset` - the start offset in the data
- `length` - the number of characters to be drawn
- `x` - the x coordinate of the anchor point
- `y` - the y coordinate of the anchor point
- `anchor` - the anchor point for positioning the text; see anchor points

```java
static final void DrawImage ( javax.microedition.lcdui.Image img, int x, int y, int anchor )
```
Draws the specified image by using the anchor point.

The image can be drawn in different positions relative to the anchor point by passing the appropriate position constants. See anchor points.

If the source image contains transparent pixels, the corresponding pixels in the destination image must be left untouched. If the source image contains partially transparent pixels, a compositing operation must be performed with the destination pixels, leaving all pixels of the destination image fully opaque.

If img is the same as the destination of this Graphics object, the result is undefined. For copying areas within an Image, copyArea should be used instead.

Parameters:

- **img** - the specified image to be drawn
- **x** - the x coordinate of the anchor point
- **y** - the y coordinate of the anchor point
- **anchor** - the anchor point for positioning the image

```java
static final void DrawLine ( int x1,
                   int y1,
                   int x2,
                   int y2
               ) [static, package, inherited]
```

Draws a line between the coordinates (x1,y1) and (x2,y2) using the current color and stroke style.

Parameters:

- **x1** - the x coordinate of the start of the line
- **y1** - the y coordinate of the start of the line
- **x2** - the x coordinate of the end of the line
- **y2** - the y coordinate of the end of the line

```java
static final void drawPartialRGB ( Graphics g,
```
Same as `drawPartialRGB` but it's useful when you draw in a back buffer that is bigger than the screen.

**Parameters:**

- `g` - graphic context, all rendering operation will occurs on this context.
- `screenWidth` - the width of the backbuffer
- `screenHeight` - the height of the backbuffer
- `rgbData` - an array of ARGB values in the format 0xARRGGBB
- `scanlength` - the relative array offset between the corresponding pixels in consecutive rows in the `rgbData` array
- `srcX` - the horizontal location of the source to be rendered
- `srcY` - the vertical location of the source to be rendered
- `x` - the horizontal location of the destination
- `y` - the vertical location of the destination
- `width` - the width of the region to be rendered
- `height` - the height of the region to be rendered
- `processAlpha` - true if `rgbData` has an alpha channel, false if all pixels are fully opaque
Same as drawRGB but it draws safely (without drawing outside the screen).

Parameters:

- $g$ - graphic context, all rendering operation will occurs on this context.
- $rgbData$ - an array of ARGB values in the format 0xAARRGGBB
- $scanlength$ - the relative array offset between the corresponding pixels in consecutive rows in the $rgbData$ array
- $srcX$ - the horizontal location of the source to be rendered
- $srcY$ - the vertical location of the source to be rendered
- $x$ - the horizontal location of the destination
- $y$ - the vertical location of the destination
- $width$ - the width of the region to be rendered
- $height$ - the height of the region to be rendered
- $processAlpha$ - true if $rgbData$ has an alpha channel, false if all pixels are fully opaque

Draws the outline of the specified rectangle using the current color and stroke style.

The resulting rectangle will cover an area $(width + 1)$ pixels wide by $(height + 1)$ pixels tall. If either width or height is less than zero, nothing is drawn.

Parameters:

- $x$ - the $x$ coordinate of the rectangle to be drawn
- $y$ - the $y$ coordinate of the rectangle to be drawn
- $width$ - the width of the rectangle to be drawn
- $height$ - the height of the rectangle to be drawn
Copies a region of the specified source image to a location within the destination, possibly transforming (rotating and reflecting) the image data using the chosen transform function.

The destination, if it is an image, must not be the same image as the source image. If it is, an exception is thrown. This restriction is present in order to avoid ill-defined behaviors that might occur if overlapped, transformed copies were permitted.

The transform function used must be one of the following, as defined in the Sprite class:

Sprite.TRANS_NONE - causes the specified image region to be copied unchanged
Sprite.TRANS_ROT90 - causes the specified image region to be rotated clockwise by 90 degrees.
Sprite.TRANS_ROT180 - causes the specified image region to be rotated clockwise by 180 degrees.
Sprite.TRANS_ROT270 - causes the specified image region to be rotated clockwise by 270 degrees.
Sprite.TRANS_MIRROR - causes the specified image region to be reflected about its vertical center.
Sprite.TRANS_MIRROR_ROT90 - causes the specified image region to be reflected about its vertical center and then rotated clockwise by 90 degrees.
Sprite.TRANS_MIRROR_ROT180 - causes the specified image region to be reflected about its vertical center and then rotated clockwise by 180 degrees.
Sprite.TRANS_MIRROR_ROT270 - causes the specified image region to be reflected about its vertical center and then rotated clockwise by 270 degrees.
If the source region contains transparent pixels, the corresponding pixels in the destination region must be left untouched. If the source region contains partially transparent pixels, a compositing operation must be performed with the destination pixels, leaving all pixels of the destination region fully opaque.

The \((x_{\text{src}}, y_{\text{src}})\) coordinates are relative to the upper left corner of the source image. The \(x_{\text{src}}, y_{\text{src}}, \text{width}, \text{height}\) parameters specify a rectangular region of the source image. It is illegal for this region to extend beyond the bounds of the source image. This requires that:
\[
x_{\text{src}} \geq 0 \\
y_{\text{src}} \geq 0 \\
x_{\text{src}} + \text{width} \leq \text{source width} \\
y_{\text{src}} + \text{height} \leq \text{source height}
\]

The \((x_{\text{dest}}, y_{\text{dest}})\) coordinates are relative to the coordinate system of this Graphics object. It is legal for the destination area to extend beyond the bounds of the Graphics object. Pixels outside of the bounds of the Graphics object will not be drawn.

The transform is applied to the image data from the region of the source image, and the result is rendered with its anchor point positioned at location \((x_{\text{dest}}, y_{\text{dest}})\) in the destination.

**Parameters:**
- \(src\): the source image to copy from
- \(x_{\text{src}}\): the \(x\) coordinate of the upper left corner of the region within the source image to copy
- \(y_{\text{src}}\): the \(y\) coordinate of the upper left corner of the region within the source image to copy
- \(width\): the width of the region to copy
- \(height\): the height of the region to copy
- \(transform\): the desired transformation for the selected region being copied
- \(x_{\text{dest}}\): the \(x\) coordinate of the anchor point in the destination drawing area
- \(y_{\text{dest}}\): the \(y\) coordinate of the anchor point in the destination drawing area
- \(anchor\): the anchor point for positioning the region within the destination image

```java
static final void DrawRGB ( int[] rgbData, ...
```
renders a series of device-independent RGB+transparency values in a specified region.

The values are stored in rgbData in a format with 24 bits of RGB and an eight-bit alpha value (0xAARRGGBB), with the first value stored at the specified offset. The scanlength specifies the relative offset within the array between the corresponding pixels of consecutive rows. Any value for scanlength is acceptable (even negative values) provided that all resulting references are within the bounds of the rgbData array. The ARGB data is rasterized horizontally from left to right within each row. The ARGB values are rendered in the region specified by x, y, width and height, and the operation is subject to the current clip region and translation for this Graphics object.

Consider P(a,b) to be the value of the pixel located at column a and row b of the Image, where rows and columns are numbered downward from the top starting at zero, and columns are numbered rightward from the left starting at zero. This operation can then be defined as:
P(a, b) = rgbData[offset + (a - x) + (b - y) * scanlength]
for
x <= a < x + width
y <= b < y + height

This capability is provided in the Graphics class so that it can be used to render both to the screen and to offscreen Image objects. The ability to retrieve ARGB values is provided by the Image.getRGB(int[], int, int, int, int, int, int, int) method.

If processAlpha is true, the high-order byte of the ARGB format specifies opacity; that is, 0x00RRGGBB specifies a fully transparent pixel and 0xFFRRGGBB specifies a fully opaque pixel. Intermediate
alpha values specify semitransparency. If the implementation does not support alpha blending for image rendering operations, it must remove any semitransparency from the source data prior to performing any rendering. (See Alpha Processing for further discussion.) If processAlpha is false, the alpha values are ignored and all pixels must be treated as completely opaque.

The mapping from ARGB values to the device-dependent pixels is platform-specific and may require significant computation.

**Parameters:**

- `rgbData` - an array of ARGB values in the format 0xAARRGGBB
- `offset` - the array index of the first ARGB value
- `scanlength` - the relative array offset between the corresponding pixels in consecutive rows in the `rgbData` array
- `x` - the horizontal location of the region to be rendered
- `y` - the vertical location of the region to be rendered
- `width` - the width of the region to be rendered
- `height` - the height of the region to be rendered
- `processAlpha` - true if `rgbData` has an alpha channel, false if all pixels are fully opaque

```java
static final void DrawRoundRect ( int x,
        int y,
        int width,
        int height,
        int arcWidth,
        int arcHeight
    )
```

Draws the outline of the specified rounded corner rectangle using the current color and stroke style.

The resulting rectangle will cover an area (width + 1) pixels wide by (height + 1) pixels tall. If either width or height is less than zero, nothing is drawn.

**Parameters:**

- `x` - the x coordinate of the rectangle to be drawn
- `y` - the y coordinate of the rectangle to be drawn
- `width` - the width of the rectangle to be drawn
The height of the rectangle to be drawn

Arc width - the horizontal diameter of the arc at the four corners

Arc height - the vertical diameter of the arc at the four corners

static final void DrawString ( String str,
                          int x,
                          int y,
                          int anchor
                      )
                      [static, package, inherited]

Draws the specified String using the current font and color.

The x,y position is the position of the anchor point

Parameters:

str - the String to be drawn
x - the x coordinate of the anchor point
y - the y coordinate of the anchor point
anchor - the anchor point for positioning the text

static final void DrawSubstring ( String str,
                                   int offset,
                                   int len,
                                   int x,
                                   int y,
                                   int anchor
                               )
                               [static, package, inherited]

Draws the specified String using the current font and color.

The x,y position is the position of the anchor point. See anchor points.
The offset and len parameters must specify a valid range of characters
within the string str. The offset parameter must be within the range [0..(str.length())], inclusive. The len parameter must be a non-negative
integer such that (offset + len) <= str.length().

Parameters:

str - the String to be drawn
static final void FillArc ( int x,
    int y,
    int width,
    int height,
    int startAngle,
    int arcAngle
)  

[static, package, inherited]  

Fills a circular or elliptical arc covering the specified rectangle.  

The resulting arc begins at startAngle and extends for arcAngle degrees. Angles are interpreted such that 0 degrees is at the 3 o'clock position. A positive value indicates a counter-clockwise rotation while a negative value indicates a clockwise rotation.  

The center of the arc is the center of the rectangle whose origin is (x, y) and whose size is specified by the width and height arguments.  

If either width or height is zero or less, nothing is drawn.  
The filled region consists of the "pie wedge" region bounded by the arc segment as if drawn by drawArc(), the radius extending from the center to this arc at startAngle degrees, and radius extending from the center to this arc at startAngle + arcAngle degrees.  

The angles are specified relative to the non-square extents of the bounding rectangle such that 45 degrees always falls on the line from the center of the ellipse to the upper right corner of the bounding rectangle. As a result, if the bounding rectangle is noticeably longer in one axis than the other, the angles to the start and end of the arc segment will be skewed farther along the longer axis of the bounds.  

Parameters:  
    x - the x coordinate of the upper-left corner of the arc to be filled.
\[
\begin{array}{l}
y \quad - \text{the y coordinate of the upper-left corner of the arc to be filled.} \\
width \quad - \text{the width of the arc to be filled} \\
height \quad - \text{the height of the arc to be filled} \\
startAngle \quad - \text{the beginning angle.} \\
arcAngle \quad - \text{the angular extent of the arc, relative to the start angle.}
\end{array}
\]

static final void FillRect ( int x,
int y,
int width,
int height
)
{ [static, package, inherited]
Fills the specified rectangle with the current color.
If either width or height is zero or less, nothing is drawn.

Parameters:
\[
\begin{align*}
x & \quad - \text{the x coordinate of the rectangle to be filled} \\
y & \quad - \text{the y coordinate of the rectangle to be filled} \\
width & \quad - \text{the width of the rectangle to be filled} \\
height & \quad - \text{the height of the rectangle to be filled}
\end{align*}
\]

static final void FillRoundRect ( int x,
int y,
int width,
int height,
int arcWidth,
int arcHeight
)
{ [static, package, inherited]
Fills the specified rounded corner rectangle with the current color.
If either width or height is zero or less, nothing is drawn.

Parameters:
\[
\begin{align*}
x & \quad - \text{the x coordinate of the rectangle to be filled} \\
y & \quad - \text{the y coordinate of the rectangle to be filled} \\
width & \quad - \text{the width of the rectangle to be filled} \\
height & \quad - \text{the height of the rectangle to be filled}
\end{align*}
\]
arcWidth - the horizontal diameter of the arc at the four corners
arcHeight - the vertical diameter of the arc at the four corners

```java
static final void FillTriangle ( int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3
)
```

Fills the specified triangle will the current color.

The lines connecting each pair of points are included in the filled triangle.

**Parameters:**

- x1 - the x coordinate of the first vertex of the triangle
- y1 - the y coordinate of the first vertex of the triangle
- x2 - the x coordinate of the second vertex of the triangle
- y2 - the y coordinate of the second vertex of the triangle
- x3 - the x coordinate of the third vertex of the triangle
- y3 - the y coordinate of the third vertex of the triangle

```java
static final void Gc ( )
```

call the garbage collector to free memory

**Note:**

gc is not guaranteed to be called immediately in order to bufferise gc call

```java
static final int GetBlueComponent ( )
```

Gets the blue component of the current color.

**Returns:**
integer value in range 0-255

```java
static final int GetClipHeight() [static, package, inherited]

Gets the height of the current clipping area.

**Returns:**
height of the current clipping area.
```

```java
static final int GetClipWidth() [static, package, inherited]

Gets the width of the current clipping area.

**Returns:**
width of the current clipping area.
```

```java
static final int GetClipX() [static, package, inherited]

Gets the X offset of the current clipping area, relative to the coordinate system origin of this graphics context.

Separating the getClip operation into two methods returning integers is more performance and memory efficient than one getClip() call returning an object.

**Returns:**
X offset of the current clipping area
```

```java
static final int GetClipY() [static, package, inherited]

Gets the Y offset of the current clipping area, relative to the coordinate system origin of this graphics context.

Separating the getClip operation into two methods returning integers is more performance and memory efficient than one getClip() call
returning an object.

**Returns:**
Y offset of the current clipping area

```java
static final int GetColor()
```

Gets the current color.

**Returns:**
an integer in form 0x00RRGGBB

```java
static final int GetDisplayColor(int color)
```

Gets the color that will be displayed if the specified color is requested.

This method enables the developer to check the manner in which RGB values are mapped to the set of distinct colors that the device can actually display. For example, with a monochrome device, this method will return either 0xFFFFFF (white) or 0x000000 (black) depending on the brightness of the specified color.

**Parameters:**

* color - the desired color (in 0x00RRGGBB format, the high-order byte is ignored)

**Returns:**
the corresponding color that will be displayed on the device's screen (in 0x00RRGGBB format)

```java
static final javax.microedition.lcdui.Font GetFont()
```

Gets the current font.

**Returns:**
current font
**static final long GetFrameTime ( )** [static, package, inherited]

get time for this frame (real time when this frame started)

**Returns:**

time

**static final int GetGrayScale ( )** [static, package, inherited]

Gets the current grayscale value of the color being used for rendering operations.

Gets the current grayscale value of the color being used for rendering operations. If the color was set by setGrayScale(), that value is simply returned. If the color was set by one of the methods that allows setting of the red, green, and blue components, the value returned is computed from the RGB color components (possibly in a device-specific fashion) that best approximates the brightness of that color.

**Returns:**

integer value in range 0-255

**static final int GetGreenComponent ( )** [static, package, inherited]

Gets the green component of the current color.

**Returns:**

integer value in range 0-255

**static final long GetRealTime ( )** [static, package, inherited]

get real time at the moment the function is called

**Returns:**

real time
**static final int GetRedComponent ( )** [static, package, inherited]

Gets the red component of the current color.

**Returns:**
integer value in range 0-255

**static final int GetScreenHeight ( )** [static, package, inherited]

get screen height, (if screen orientation changes, it will return the new screen height automatically

**Returns:**
screen height

**static final int GetScreenWidth ( )** [static, package, inherited]

get screen width, (if screen orientation changes, it will return the new screen width automatically

**Returns:**
screen width

**static final int GetStrokeStyle ( )** [static, package, inherited]

Gets the stroke style used for drawing operations.

**Returns:**
stroke style, SOLID or DOTTED

**static final int GetTranslateX ( )** [static, package, inherited]

Gets the X coordinate of the translated origin of this graphics context.

**Returns:**
X of current origin

```java
static final int GetTranslateY() { [static, package, inherited]

    // Gets the Y coordinate of the translated origin of this graphics context.

    **Returns:**
    Y of current origin

    PlatformRequest(String url) { [static, package, inherited]

    // Performs platformRequest.
    // This method wraps calls to platformRequest, incorporating some workarounds for usual platformRequest problems.

    **Parameters:**
    url the url to send to the browser

    **Returns:**
    if the platformRequest was successful

    SetClip(int x, int y, int width, int height) { [static, package, inherited]

    // Sets the current clip to the rectangle specified by the given coordinates.
    // Rendering operations have no effect outside of the clipping area.

    **Parameters:**
    x - the x coordinate of the new clip rectangle
    y - the y coordinate of the new clip rectangle
    width - the width of the new clip rectangle
```
*height* - the height of the new clip rectangle

```java
static final void setColor ( int red,
   int green,
   int blue
 )
```

Sets the current color to the specified RGB values.

All subsequent rendering operations will use this specified color.

**Parameters:**

- `red` - the red component of the color being set in range 0-255
- `green` - the green component of the color being set in range 0-255
- `blue` - the blue component of the color being set in range 0-255

```java
static final void SetColor ( int RGB )
```

Sets the current color to the specified RGB values.

All subsequent rendering operations will use this specified color. The RGB value passed in is interpreted with the least significant eight bits giving the blue component, the next eight more significant bits giving the green component, and the next eight more significant bits giving the red component. That is to say, the color component is specified in the form of 0x00RRGGBB. The high order byte of this value is ignored.

**Parameters:**

- `RGB` - the color being set

```java
static final void SetCurrentGraphics ( javax.microedition.lcdui.Image img )
```

set current graphics context -.

all rendering operation will occurs on this context

**Parameters:**
get the current graphic context from this image, or null to reset

**Note:**

graphic context is always reseted at beginning of each frame

```java
static final void SetCurrentGraphics ( javax.microedition.lcdui.Graphics graphics ) {
    set current graphics context -.
    all rendering operation will occurs on this context

    **Parameters:**

    *graphics* - graphic context, all rendering operation will occurs on this context, or null to reset

    **Note:**

graphic context is always reseted at beginning of each frame
```

```java
static final void SetFont ( javax.microedition.lcdui.Font font ) {
    Sets the font for all subsequent text rendering operations.
    If font is null, it is equivalent to setFont(Font.getDefaultFont()).

    **Parameters:**

    *font* - the specified font
```

```java
static final void SetGrayScale ( int value ) {
    Sets the current grayscale to be used for all subsequent rendering operations.
    For monochrome displays, the behavior is clear. For color displays, this
    sets the color for all subsequent drawing operations to be a gray color
    equivalent to the value passed in. The value must be in the range 0-255.
```
**Parameters:**

*value* - the desired grayscale value

---

```java
static final void SetStrokeStyle ( int style )
```

Sets the stroke style used for drawing lines, arcs, rectangles, and rounded rectangles.

This does not affect fill, text, and image operations.

**Parameters:**

*style* - can be SOLID or DOTTED

---

```java
static final void Translate ( int x, int y )
```

Translates the origin of the graphics context to the point (x, y) in the current coordinate system.

Translates the origin of the graphics context to the point (x, y) in the current coordinate system. All coordinates used in subsequent rendering operations on this graphics context will be relative to this new origin.

The effect of calls to translate() are cumulative. For example, calling translate(1, 2) and then translate(3, 4) results in a translation of (4, 6).

The application can set an absolute origin (ax, ay) using the following technique:

```
g.translate(ax - g.getTranslateX(), ay - g.getTranslateY())
```

**Parameters:**

*x* - the x coordinate of the new translation origin

*y* - the y coordinate of the new translation origin
static void Vibrate (int duration) [static, inherited]

Make the phone vibrate or flash the back light is the phone has this functionnality.

Parameters:
  duration - duration of the vibration (or backlight flash)
## Variable Documentation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>final int BOTTOM</td>
<td>32</td>
<td>Constant for positioning the anchor point of text and images below the text or image.</td>
</tr>
<tr>
<td>final int DOTTED</td>
<td>1</td>
<td>Constant for the DOTTED stroke style.</td>
</tr>
<tr>
<td>final int HCENTER</td>
<td>1</td>
<td>Constant for centering text and images horizontally around the anchor point.</td>
</tr>
<tr>
<td>final int LEFT</td>
<td>4</td>
<td>Constant for positioning the anchor point of text and images to the left of the text or image.</td>
</tr>
<tr>
<td>final int RIGHT</td>
<td>8</td>
<td>Constant for positioning the anchor point of text and images to the right of the text or image.</td>
</tr>
<tr>
<td>final int SOLID</td>
<td>0</td>
<td>Constant for the SOLID stroke style.</td>
</tr>
</tbody>
</table>
**final int TOP = 16** [static, package, inherited]

Constant for positioning the anchor point of text and images above the text or image.

**final int TRANS_MIRROR = 2** [static, package, inherited]

causes the specified image region to be reflected about its vertical center.

**final int TRANS_MIRROR_ROT180 = 1** [static, package, inherited]

causes the specified image region to be reflected about its vertical center and then rotated clockwise by 180 degrees.

**final int TRANS_MIRROR_ROT270 = 4** [static, package, inherited]

causes the specified image region to be reflected about its vertical center and then rotated clockwise by 270 degrees.

**final int TRANS_MIRROR_ROT90 = 7** [static, package, inherited]

causes the specified image region to be reflected about its vertical center and then rotated clockwise by 90 degrees.

**final int TRANS_NONE = 0** [static, package, inherited]

No transform is applied.

**final int TRANS_ROT180 = 3** [static, package, inherited]

causes the specified image region to be rotated clockwise by 180
degrees.

### final int TRANS_ROT270 = 6 [static, package, inherited]

causes the specified image region to be rotated clockwise by 270 degrees.

### final int TRANS_ROT90 = 5 [static, package, inherited]

causes the specified image region to be rotated clockwise by 90 degrees.

### final int VCENTER = 2 [static, package, inherited]

Constant for centering text and images vertically around the anchor point.
GLLib : memory

[GLLib]

memory access function (byte array manipulation) More...
## Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static void Mem_ArrayCopy(Object src, int src_position, Object dst, int dst_position, int length)</code> throws Exception</td>
<td>Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array.</td>
</tr>
<tr>
<td><code>static int Mem_GetArray(byte[] src, int src_off, byte[] dst)</code> throws Exception</td>
<td>fill destination array with content of src array at specified offset</td>
</tr>
<tr>
<td><code>static byte Mem_GetByte(byte[] src, int src_off)</code></td>
<td>get a byte value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>static int Mem_GetInt(byte[] src, int src_off)</code></td>
<td>get an int value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>static long Mem_GetLong(byte[] src, int src_off)</code></td>
<td>get a long value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>static short Mem_GetShort(byte[] src, int src_off)</code></td>
<td>get a short value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>static Object Mem_ReadArray(InputStream is)</code></td>
<td>Read an array/multiarray, from an inputStream.</td>
</tr>
<tr>
<td><code>static int Mem_SetArray(byte[] dst, int dst_off, byte[] src)</code> throws Exception</td>
<td>copies whole content of src array in dst array at specified offset</td>
</tr>
<tr>
<td><code>static int Mem_SetByte(byte[] dst, int dst_off, byte src)</code></td>
<td>set a byte value in a byte array at a given offset</td>
</tr>
<tr>
<td><code>static int Mem_SetInt(byte[] dst, int dst_off, int src)</code></td>
<td>set a int value in a byte array at a given offset</td>
</tr>
<tr>
<td><code>static int Mem_SetLong(byte[] dst, int dst_off, long src)</code></td>
<td>set a long value in a byte array at a given offset</td>
</tr>
<tr>
<td><code>static int Mem_SetShort(byte[] dst, int dst_off, short src)</code></td>
<td>set a short value in a byte array at a given offset</td>
</tr>
<tr>
<td><code>static int Stream_Read(InputStream is)</code> throws Exception</td>
<td></td>
</tr>
<tr>
<td><code>static int Stream_Read16(InputStream is)</code> throws Exception</td>
<td></td>
</tr>
<tr>
<td><code>static int Stream_Read32(InputStream is)</code> throws Exception</td>
<td></td>
</tr>
<tr>
<td><code>static int Stream_ReadFully(InputStream is, byte[] array, int offset, int length)</code></td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description
memory access function (byte array manipulation)
Function Documentation

```java
static void Mem_ArrayCopy ( Object src,
   int src_position,
   Object dst,
   int dst_position,
   int length )
   throws Exception
```

Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array.

**Parameters:**
- `src` - the source array.
- `src_position` - start position in the source array.
- `dst` - the destination array.
- `dst_position` - pos start position in the destination data.
- `length` - the number of array elements to be copied.

```java
static int Mem_GetArray ( byte[] src,
                           int src_off,
                           byte[] dst )
   throws Exception
```

fill destination array with content of src array at specified offset

**Parameters:**
- `src` - byte array to copy values from
- `src_off` - offset in the source array to find values at
- `dst` - destination byte array

**Returns:**
new src offset value

```java
static byte Mem_GetByte ( byte[] src,
                          int src_off )
```

[static, package, inherited]
get a byte value from the specified array at specified offset

**Parameters:**

- `src` - byte array to read value from
- `src_off` - offset in the byte array

**Returns:**

byte value

```c
static int Mem_GetInt ( byte[] src, int src_off )
```

get an int value from the specified array at specified offset

**Parameters:**

- `src` - byte array to read value from
- `src_off` - offset in the byte array

**Returns:**

int value

```c
static int Mem_GetInt ( byte[] src, int src_off )
```

get a long value from the specified array at specified offset

**Parameters:**

- `src` - byte array to read value from
- `src_off` - offset in the byte array

**Returns:**

long value

```c
static long Mem_GetLong ( byte[] src, int src_off )
```
get a short value from the specified array at specified offset

**Parameters:**
- `src` - byte array to read value from
- `src_off` - offset in the byte array

**Returns:**
- short value

---

```java
static Object Mem_ReadArray ( InputStream is )
```

Read an array/multiarray, from an inputStream.

**Parameters:**
- `is` inputstream to read array from

**Returns:**
- array or multiarray as an Object.

**Exceptions:**
- `Exception` if error occurred.

---

```java
static int Mem_SetArray ( byte[] dst, int dst_off, byte[] src )
```

copies whole content of `src` array in `dst` array at specified offset

**Parameters:**
- `dst` - byte array to modify
- `dst_off` - offset in the destination byte array array
- `src` - source byte array

**Returns:**
- new offset value
static int Mem_SetByte ( byte[]  dst,
    int  dst_off,
    byte src ) [static, package, inherited]

set a byte value in a byte array at a given offset

Parameters:

  dst  - byte array to modify
  dst_off  - offset in the byte array to modify
  src  - byte value to set in the byte array

Returns:

  new offset value

static int Mem_SetInt ( byte[]  dst,
    int  dst_off,
    int  src ) [static, package, inherited]

set a int value in a byte array at a given offset

Parameters:

  dst  - byte array to modify
  dst_off  - offset in the byte array to modify
  src  - int value to set in the byte array

Returns:

  new offset value

static int Mem_SetLong ( byte[]  dst,
    int  dst_off,
    long src ) [static, package, inherited]

set a long value in a byte array at a given offset
Parameters:

- `dst` - byte array to modify
- `dst_off` - offset in the byte array to modify
- `src` - long value to set in the byte array

Returns:

ew offset value

```
static int Mem_SetShort ( byte[] dst,
            int dst_off,
            short src )
```

set a short value in a byte array at a given offset

Parameters:

- `dst` - byte array to modify
- `dst_off` - offset in the byte array to modify
- `src` - short value to set in the byte array

Returns:

ew offset value

```
static int Stream_Read ( InputStream is )
```
GLLib : text

[GLLib]

text decoding and caching function More...
Modules

GLLang
### Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static void Text_BuildStringCache()</code></td>
<td>Cache all the text into string Array</td>
</tr>
<tr>
<td><code>static void Text_FreeAll()</code></td>
<td>Free all the data-array used by the text</td>
</tr>
<tr>
<td><code>static String Text_FromUTF8(byte[] src, int offset, int len)</code></td>
<td>Get a string from an UTF-8 encoded byte array.</td>
</tr>
<tr>
<td><code>String Text_GetLanguageAsString(int languageCode)</code></td>
<td>Get language code as a string</td>
</tr>
<tr>
<td><code>static final int Text_GetNbString()</code></td>
<td>Get nb of string in current text pack</td>
</tr>
<tr>
<td><code>int Text_GetPhoneDefaultLangage()</code></td>
<td>Get phone default language, if unable to get it, return <code>GLLang.EN</code> (considered as default language)</td>
</tr>
<tr>
<td><code>static String Text_GetString(int index)</code></td>
<td>Get a string given it's index</td>
</tr>
<tr>
<td><code>static void Text_LoadTextFromPack(String filename, int index1, int index2)</code></td>
<td>Open and load text from two text packages inside a pack file</td>
</tr>
<tr>
<td><code>static void Text_LoadTextFromPack(String filename, int index)</code></td>
<td></td>
</tr>
<tr>
<td><code>static final void Text_SetEncoding(String encoding)</code></td>
<td>Set current text encoding</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>text_encoding = &quot;UTF-8&quot;</td>
<td>encoding for text (UTF-8 by default)</td>
</tr>
<tr>
<td>static int</td>
<td>text_nbString</td>
<td>number of string in this text pack</td>
</tr>
</tbody>
</table>
Detailed Description

text decoding and caching function
Function Documentation

static void Text_BuildStringCache ( ) [static, package, inherited]

- cache all the text into string Array
- using this function will store the text in String Array instead of byte Array
- byte Array consume less memory, but must allocate a String each time it's needed
- String Array consume more heap memory, but only allocate one string once and for all for each text

static void Text_FreeAll ( ) [static, package, inherited]

- free all the data-array used by the text

static String Text_FromUTF8 ( byte[] src, int offset, int len ) [static, package, inherited]

- get a string from an UTF-8 encoded byte array.
- Text_GetString will use this function if the string is encoded with UTF-8 and if the flag text_useInternalUTF8Converter is set.

Parameters:
- src UTF-8 encoded byte array.
- offset Offset to the start of the string in the src array.
- len Length of the string to convert.

Returns:
- a String

Note:
This function is slow, used the internal String() conversion if possible.

**String Text_GetLanguageAsString ( int languageCode )** [package, inherited]

get language code as a string

**Parameters:**

*languageCode* language code to get as a string

**Returns:**

language code as string, or null if invalid language code

**static final int Text_GetNbString ( )** [static, package, inherited]

get nb of string in current text pack

**Returns:**

nb of string in current text pack

**int Text_GetPhoneDefaultLangage ( )** [package, inherited]

get phone default langage, if unable to get it, return **GLLang.EN** (considered as default language)

**Returns:**

phone default language as defined in **GLLang** interface

**See also:**

**GLLang**

**static String Text_GetString ( int index )** [static, package, inherited]

get a string given it's index
Parameters:

   index  index of the string to get

Returns:

   a string

static void Text_LoadTextFromPack ( String filename,
   int   index1,
   int   index2
 )  [static, package, inherited]

open and load text from two text packages inside a pack file

Parameters:

   filename  name of the pack file
   index1    index of the first text package to load
   index2    index of the second text package to load

Note:

   this function will call automatically Text_BuildStringCache() if
   GLLibConfig.text_useStringCache is set to true

See also:

   GLLibConfig.text_useStringCache
   Text_BuildStringCache

static void Text_LoadTextFromPack ( String filename,
   int   index
 )  [static, package, inherited]

static final void Text_SetEncoding ( String encoding )  [static, package, inherited]

set current text encoding

Parameters:

   encoding  encoding of the text
### Variable Documentation

<table>
<thead>
<tr>
<th>String text_encoding = &quot;UTF-8&quot; [static, package, inherited]</th>
</tr>
</thead>
<tbody>
<tr>
<td>encoding for text (UTF-8 by default)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int text_nbString [static, package, inherited]</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of string in this text pack</td>
</tr>
</tbody>
</table>
GLLang
[GLLib : text]
## Classes

<table>
<thead>
<tr>
<th>interface</th>
<th>GLLang</th>
</tr>
</thead>
<tbody>
<tr>
<td>language code</td>
<td>More...</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLLib by [doxygen](https://www.doxygen.org/) 1.5.2
GLLib : Recordstore management

[GLLib]

handle RMS (save data) access (read write) More...
Modules

<table>
<thead>
<tr>
<th>GLLib</th>
<th>GLLib Profiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility functions for profiling in emulator and phone.</td>
<td></td>
</tr>
</tbody>
</table>
## Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static void InitSharedRms</td>
<td>(String strVendor, String strMidletName)</td>
<td>Sets vendor name and midlet names - used for shared rms read/write.</td>
</tr>
<tr>
<td>static void Pack_CloseShared</td>
<td>() throws Exception</td>
<td>Closes current pack, previously opened with <code>Pack_OpenShared</code>.</td>
</tr>
<tr>
<td>static void Pack_OpenShared</td>
<td>(String strName, String strVendor, String strMidletName) throws Exception</td>
<td>Opens a shared rms pack.</td>
</tr>
<tr>
<td>static byte[] Rms_Read</td>
<td>(String strName)</td>
<td>Reads data from the recordstore.</td>
</tr>
<tr>
<td>static byte[] Rms_ReadShared</td>
<td>(String strName, String strVendor, String strMidletName)</td>
<td>Read data from shared recordstore.</td>
</tr>
<tr>
<td>static void Rms_Write</td>
<td>(String strName, byte[] data)</td>
<td>Writes data to the recordstore.</td>
</tr>
<tr>
<td>static void Rms_WriteShared</td>
<td>(String strName, String strVendor, String strMidletName, byte[] data)</td>
<td>Write data to shared recordstore.</td>
</tr>
<tr>
<td>static void SavePack</td>
<td>(String packName, String rmsName)</td>
<td>Reads a resource, and writes it into the recordstore - used for DEBUG purposes.</td>
</tr>
</tbody>
</table>
Detailed Description

handle RMS (save data) access (read write)
Function Documentation

```java
static void InitSharedRms ( String strVendor,
    String strMidletName )
```

Sets vendor name and midlet names - used for shared rms read/write.

**Note:**
Use only if `GLLibConfig.rms_useSharing` == true

**Parameters:**
- `strVendor` Value of the field MIDlet-Vendor in jad/manifest of the application which is the owner of the rms
- `strMidletName` Value of the field MIDlet-Name in jad/manifest of the application which is the owner of the rms

```java
static void Pack_CloseShared ( ) throws Exception
```

Closes current pack, previously opened with Pack_OpenShared.

**Note:**
Use only if `GLLibConfig.rms_useSharing` == true and `GLLibConfig.rms_usePackRead` == true

```java
static void Pack_OpenShared ( String strName,
    String strVendor,
    String strMidletName )
```  

Opens a shared rms pack.

**Note:**
Use only if `GLLibConfig.rms_useSharing` == true and `GLLibConfig.rms_usePackRead` == true

**Parameters:**
static byte [] Rms_Read ( String \textit{strName} ) \hspace{1em} \text{[static, package, inherited]}

Reads data from the recordstore.

**Parameters:**
- \textit{strName} Filename of recordstore to read from

**Returns:**
A byte array, containing the recordstore data

static byte [] Rms_ReadShared ( String \textit{strName},
String \textit{strVendor},
String \textit{strMidletName} ) \hspace{1em} \text{[static, package, inherited]}

Read data from shared recordstore.

**Note:**
Use only if \texttt{GLLibConfig.rms\_useSharing == true}

**Parameters:**
- \textit{strName} Rms name
- \textit{strVendor} Value of the field MIDlet-Vendor in jad/manifest of the application which is the owner of the rms
- \textit{strMidletName} Value of the field MIDlet-Name in jad/manifest of the application which is the owner of the rms

static void Rms_Write ( String \textit{strName},
byte[] \textit{data} ) \hspace{1em} \text{[static, package, inherited]}

Writes data to the recordstore.
Parameters:

strName  Filename of recordstore to write to

strVendor  Value of the field MIDlet-Vendor in jad/manifest of the application which is the owner of the rms

strMidletName  Value of the field MIDlet-Name in jad/manifest of the application which is the owner of the rms

data  Byte array data to write in the rms

static void Rms_WriteShared ( String strName, String strVendor, String strMidletName, byte[] data )

Write data to shared recordstore.

Note:

Use only if GLLibConfig.rms_useSharing == true

Parameters:

packName  Filename of a pack to read from

rmsName  Name of a recordstore to write to

static void SavePack ( String packName, String rmsName )

Reads a resource, and writes it into the the recordstore - used for DEBUG purposes.
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
</table>


GLLib : GLLib Profiler

[GLLib : Recordstore management, GLLib]

Utility functions for profiling in emulator and phone. More...
Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profiler_BeginNamedEvent</strong> (String name)</td>
<td>Mark the beginning of an event.</td>
</tr>
<tr>
<td><strong>Profiler_Draw</strong> ()</td>
<td>Draw profiler events to screen.</td>
</tr>
<tr>
<td><strong>Profiler_End</strong> ()</td>
<td>End capturing events.</td>
</tr>
<tr>
<td><strong>Profiler_EndNamedEvent</strong> ()</td>
<td>Mark the end of an event.</td>
</tr>
<tr>
<td><strong>Profiler_Start</strong> ()</td>
<td>Start capturing events.</td>
</tr>
</tbody>
</table>
## Variables

<table>
<thead>
<tr>
<th>static final int</th>
<th>PROFILER_MAX_EVENTS = 200</th>
</tr>
</thead>
</table>

*Maximum number of events.*
Detailed Description

Utility functions for profiling in emulator and phone.
Function Documentation

**static void** Profiler_BeginNamedEvent ( String *name*) [static, package, inherited]

Mark the beginning of an event.

**Parameters:**

*name* Name of the event.

**static void** Profiler_Draw ( ) [static, package, inherited]

Draw profiler events to screen.

**static void** Profiler_End ( ) [static, package, inherited]

End capturing events.

**static void** Profiler_EndNamedEvent ( ) [static, package, inherited]

Mark the end of an event.

**static void** Profiler_Start ( ) [static, package, inherited]

Start capturing events.
final int PROFILER_MAX_EVENTS = 200 [static, package, inherited]

Maximum number of events.
GLLib Configuration file

configuration file for the gllib. More...
## Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLibConfig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GLLib setup class. <a href="#">More...</a></td>
</tr>
</tbody>
</table>
Detailed Description

can all field can be overriden using GLASM
## Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLibPathFinding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A* (A star) Pathfinding class. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLib by [doxygen](https://www.doxygen.org) 1.5.2
GLLibPlayer
## Modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLLibPlayer : AnimationPlayer</td>
<td>ASprite animation player.</td>
</tr>
<tr>
<td>GLLibPlayer : SoundPlayer</td>
<td>Sound player</td>
</tr>
<tr>
<td>GLLibPlayer : Tileset</td>
<td>Tile set loading/drawing function</td>
</tr>
</tbody>
</table>
## Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLibPlayer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Generic GLLib Player. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLLib by [doxygen](http://www.stackexchange.com) 1.5.2
GLLibPlayer : AnimationPlayer
[GLLibPlayer]

ASprite animation player. More...
### Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final int GetAnim()</code></td>
<td>get animation</td>
</tr>
<tr>
<td><code>final int GetDuration()</code></td>
<td>Get the duration of the current frame of the current animation.</td>
</tr>
<tr>
<td><code>final int GetFrame()</code></td>
<td>get current frame nb</td>
</tr>
<tr>
<td><code>int GetNbanim()</code></td>
<td>Get the animation count of the current sprite.</td>
</tr>
<tr>
<td><code>int GetNbFrame()</code></td>
<td>Get the frame count of the current animation.</td>
</tr>
<tr>
<td><code>final ASprite GetSprite()</code></td>
<td>Get the current sprite.</td>
</tr>
<tr>
<td><code>final int GetTransform()</code></td>
<td>get current sprite transformation</td>
</tr>
<tr>
<td><code>GLLibPlayer (ASprite sprite, int x, int y)</code></td>
<td>Basic constructor.</td>
</tr>
<tr>
<td><code>GLLibPlayer ()</code></td>
<td>Empty constructor.</td>
</tr>
<tr>
<td><code>boolean IsAnimOver()</code></td>
<td>Query the state of the current animation to see if its over.</td>
</tr>
<tr>
<td><code>void Render ()</code></td>
<td>Render the current animation in GLLib current graphic context.</td>
</tr>
<tr>
<td><code>void Reset ()</code></td>
<td>Reset the current player.</td>
</tr>
<tr>
<td><code>void SetAnim (int anim, int nbLoop)</code></td>
<td>Set a new animation number to play.</td>
</tr>
<tr>
<td><code>final void SetAnim (int anim)</code></td>
<td>Set a new animation number to play, animation will loop forever.</td>
</tr>
<tr>
<td><code>int SetFrame (int frame)</code></td>
<td>Set a new frame position in the current animation.</td>
</tr>
<tr>
<td><code>final void SetPos (int x, int y)</code></td>
<td>Set new position of sprite.</td>
</tr>
<tr>
<td><code>void SetSprite (ASprite sprite)</code></td>
<td>Set a new ASprite reference in the player.</td>
</tr>
<tr>
<td><code>final void SetTransform (int transform)</code></td>
<td>set current sprite transformation</td>
</tr>
<tr>
<td><code>void Update (int DT)</code></td>
<td>Update current animation time from 1 time unit.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td><code>final void Update()</code></td>
<td>Update current animation time from 1 time unit.</td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int</code></td>
<td><code>curFlags</code></td>
<td>Sprite player current display flags.</td>
</tr>
<tr>
<td><code>int</code></td>
<td><code>curTime</code></td>
<td>Sprite current time. Used for frame that stay on screen longer than a frame.</td>
</tr>
<tr>
<td><code>int</code></td>
<td><code>posX</code></td>
<td>Sprite player current X position.</td>
</tr>
<tr>
<td><code>int</code></td>
<td><code>posY</code></td>
<td>Sprite player current Y position.</td>
</tr>
<tr>
<td><code>ASprite</code></td>
<td><code>sprite</code></td>
<td>Sprite player current <code>ASprite</code> reference.</td>
</tr>
</tbody>
</table>
Detailed Description

ASprite animation player.

GLLibPlayer need to be instanciated to use those function
### Function Documentation

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| **final int GetAnim()** [package, inherited] | get animation  
**Returns:** current selected animation, -1 if none |
| **final int GetDuration()** [package, inherited] | Get the duration of the current frame of the current animation.  
**Returns:** The duration of the current frame of the current animation or 0 if the animation is not set |
| **final int GetFrame()** [package, inherited] | get current frame nb  
**Returns:** current frame nb in current animation, result is invalid if no animation is set |
| **int GetNbanim()** [package, inherited] | Get the animation count of the current sprite.  
**Returns:** The animation count of the current sprite |
| **int GetNbFrame()** [package, inherited] | |
Get the frame count of the current animation.

**Returns:**
The frame count of the current animation or -1 if the animation is not set.

```java
final ASprite GetSprite() [package, inherited]
```

Get the current sprite.

```java
final int GetTransform() [package, inherited]
```

get current sprite transformation

**Returns:**
one of GLib.TRANS_NONE, GLib.TRANS_ROT90,
GLib.TRANS_ROT180, GLib.TRANS_ROT270,
GLib.TRANS_MIRROR, GLib.TRANS_MIRROR_ROT90,
GLib.TRANS_MIRROR_ROT180,
GLib.TRANS_MIRROR_ROT270

```java
GLLibPlayer( ASprite sprite, int x, int y ) [package, inherited]
```

Basic constructor.

Default constructor to use to Play some animations.

**Parameters:**

- `sprite` Sprite to use by this player. The player will only keep a reference to the `ASprite` eg it won't be copied.
- `x` Initial X Position of the player.
- `y` Initial Y Position of the player.
### GLLibPlayer() [package, inherited]

Empty constructor.

**Note:**
It is up to user to fill all information.

### boolean IsAnimOver() [package, inherited]

Query the state of the current animation to see if its over.

**Returns:**
True if the current animation is over.

### void Render() [package, inherited]

Render the currency animation in **GLLib** current graphic context.

The animation will be drawn at posX,posY.

**See also:**
GLLib.setCurrentGraphics

### void Reset() [package, inherited]

Reset the current player.

The player will be in the same state as if you would have called the empty constructor.

**Note:**
Usefull only on the **ASprite** player, no effect on the sound.

### void SetAnim(int anim, int nbLoop)


Set a new animation number to play.

**Parameters:**

- `anim` Animation number to play
- `nbLoop` how many time this animation should loop (-1 for forever)(cannot be 0)

**Note:**

if an animation is currently playing, and try a setanim on the same anim . nothing is done, do SetAnim(-1, 1);SetAnim(anim, loop) to force an update

---

```
final void SetAnim ( int anim )
```

Set a new animation number to play, animation will loop forever.

**Parameters:**

- `anim` Animation number to play. The animation will reset the next frame to play of this new animation to 0.

**Note:**

if an animation is currently playing, and try a setanim on the same anim . nothing is done, do SetAnim(-1, 1);SetAnim(anim, loop) to force an update

**Deprecated:**

use SetAnim (int anim, int nbLoop) instead

---

```
int SetFrame ( int frame )
```

Set a new frame position in the current animation.

**Parameters:**

- `frame` New frame position.

**Returns:**

Frame adjusted if frame was larger than the frame count. Ex: if
you ask for frame 7 of a 5 frame animation, this call will place the frame at 2. return -1 if no animation

**Note:**
In DEBUG, there will be an assert if frame is smaller than 0, but not in RELEASE.

```java
final void SetPos ( int x,
    int y
 ) [package, inherited]
```

Set new position of sprite.

**Parameters:**
- `x` Sprite new X Position.
- `y` Sprite new Y Position.

**Note:**
There are no validation of the new position, it can be outside of the screen. You can also access directly the posX and posY variables.

```java
void SetSprite ( ASprite sprite ) [package, inherited]
```

Set a new `ASprite` reference in the player.

This call will also reset the animation number.

**Parameters:**
- `sprite` Sprite to be used from now on. If the sprite is null, this will remove the reference to any sprite from this player.

```java
final void SetTransform ( int transform ) [package, inherited]
```

set current sprite transformation

**Parameters:**
- `transform` any of GLLib transform
See also:

GLLib.TRANS_NONE, GLLib.TRANS_ROT90, GLLib.TRANS_ROT180, GLLib.TRANS_ROT270, GLLib.TRANS_MIRROR, GLLib.TRANS_MIRROR_ROT90, GLLib.TRANS_MIRROR_ROT180, GLLib.TRANS_MIRROR_ROT270

```java
void Update ( int DT ) [package, inherited]
```

Update current animation time from 1 time unit.

If the current frame time is exceeded the current frame will be increased, if the current frame reach the animation frame count, the current frame will be looped to 0.

**Parameters:**

DT nb of time unit elapsed since last frame

```java
final void Update ( ) [package, inherited]
```

Update current animation time from 1 time unit.

If the current frame time is exceeded the current frame will be increased, if the current frame reach the animation frame count, the current frame will be looped to 0.

**Deprecated:**

use void Update(int DT) instead
### Variable Documentation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int curFlags [package, inherited]</code></td>
<td>Sprite player current display flags.</td>
</tr>
<tr>
<td><code>int curTime [inherited]</code></td>
<td>Sprite current time. Used for frame that stay on screen longer than a frame.</td>
</tr>
<tr>
<td><code>int posX [package, inherited]</code></td>
<td>Sprite player current X position.</td>
</tr>
</tbody>
</table>
GLLibPlayer : SoundPlayer
[GLLibPlayer]

sound player More...
## Functions

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>run ()</code></td>
<td>Implementation of the <code>Runnable</code> interface.</td>
</tr>
<tr>
<td>static long</td>
<td><code>Snd_DurationGet (int channel)</code> throws Exception</td>
<td>get duration of a sound</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_ForceExecOnThreadOnGamePause ()</code></td>
<td>When the game receive an interrupt, it can be important to force execution of the sound thread.</td>
</tr>
<tr>
<td>static final void</td>
<td><code>Snd_FreeChannel (int channel)</code></td>
<td>Free sound channel from the player assigned to it.</td>
</tr>
<tr>
<td>Object</td>
<td><code>Snd_GetChannelPlayer (int channel)</code></td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td><code>Snd_GetChannelVolume (int channel)</code> throws Exception</td>
<td>Get the current volume of a channel.</td>
</tr>
<tr>
<td>static int</td>
<td><code>Snd_GetCurrentSoundIndex (int nChannel)</code></td>
<td>Get the index of the sound loaded on the channel.</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_Init (int nbSoundSlot)</code> throws Exception</td>
<td>Allocate sound player and resources.</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>Snd_IsPlaying (int channel)</code> throws Exception</td>
<td>Return true if a sound is currently playing on channel.</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_LoadSound (byte[] soundData, int nMIME, int index, boolean bCacheThisSound)</code> throws Exception</td>
<td>Load a sound file/resource from data package.</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_LoadSound (byte[] soundData, int nMIME, int index)</code> throws Exception</td>
<td>Load a sound file/resource from data package.</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_LoadSound (String dataFileName, int resourceIndex, boolean bCacheThisSound)</code> throws Exception</td>
<td>Load a sound file/resource from data package.</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_LoadSound (String dataFileName, int resourceIndex)</code> throws Exception</td>
<td>Load a sound file/resource from data package.</td>
</tr>
<tr>
<td>static long</td>
<td><code>Snd_MediaTimeGet (int channel)</code></td>
<td>get current media time</td>
</tr>
<tr>
<td>static long</td>
<td><code>Snd_MediaTimeSet (int channel, long time)</code></td>
<td>set current media time</td>
</tr>
<tr>
<td>static void</td>
<td><code>Snd_MidiPlayNote (int channel, int MIDIChannel, int note, int volume)</code> throws Exception</td>
<td>play a note on a midi channel.</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>Snd_MidiSetChannelVolume (int channel, int MIDIChannel, int volume)</code> throws Exception</td>
<td>adjust volume of a midi channel</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Pause</strong> (int channel)</td>
<td>Pause sound on a channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Play</strong> (int channel, int index, int loop, int volume, int priority)</td>
<td>Play a sound on a channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_PrepareSound</strong> (int channel, int index, int priority)</td>
<td>Prepare a sound to be played on a channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Quit</strong> () throws Exception</td>
<td>Deallocate all sound resources and players.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_RateGet</strong> (int channel)</td>
<td>Get rate of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_RateGetMax</strong> (int channel) throws Exception</td>
<td>Get max rate of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_RateGetMin</strong> (int channel) throws Exception</td>
<td>Get min rate of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_RateSet</strong> (int channel, int rate) throws Exception</td>
<td>Set rate of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Resume</strong> (int channel)</td>
<td>Resume currently paused sound on a channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_SetMasterVolume</strong> (int volume) throws Exception</td>
<td>Change master volume value.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_SetMediaDuration</strong> (int index, int duration) throws Exception</td>
<td>Sets duration for the sound (Use with GLLibConfig.sound_useFakeMediaDuration == true).</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Stop</strong> (int channel)</td>
<td>Stop sound on a channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_StopAllSounds</strong> ()</td>
<td>Stop all sounds on all channel.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_TempoGet</strong> (int channel) throws Exception</td>
<td>Get tempo of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_TempoSet</strong> (int channel, int tempo) throws Exception</td>
<td>Set tempo of a sound</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_UnLoadSound</strong> (int index) throws Exception</td>
<td>Unload a sound resource from memory.</td>
<td></td>
</tr>
<tr>
<td><strong>Snd_Update</strong> ()</td>
<td>Update sound engine status all sound resource.</td>
<td></td>
</tr>
</tbody>
</table>
### Variables

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td><strong>k_snd_priority_highest</strong> = 0</td>
<td>Sound priorities constant.</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>k_snd_priority_lowest</strong> = 15</td>
<td>Sound priorities constant.</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>k_snd_priority_normal</strong> = 7</td>
<td>Sound priorities constant.</td>
</tr>
<tr>
<td>static String[]</td>
<td><strong>s_data_mimeType</strong></td>
<td>Sound Mime Container, to know the sound type of each slots.</td>
</tr>
<tr>
<td>static int</td>
<td><strong>s_snd_masterVolume</strong></td>
<td>Sound engine master volume (0-100). The volume will be scaled to the phone range.</td>
</tr>
<tr>
<td>static int</td>
<td><strong>s_snd_maxNbSoundSlot</strong></td>
<td>Sound maximum slot count for this game. The maximum number of sounds that can be loaded into memory.</td>
</tr>
</tbody>
</table>
Detailed Description

sound player

**GLLibPlayer** does not need to be instanciated to use those function
**Function Documentation**

### void run ( ) [inherited]

Implementation of the `Runnable` interface.

This function will be used only when the `GLLibPlayer` is used to play sounds. When the `GLLibPlayer` is instanciated to play anymations, this functions is doing nothing.

### static long Snd_DurationGet ( int channel ) throws Exception [static, package, inherited]

get duration of a sound

**Parameters:**

- `channel` - channel of the sound

**Note:**

will only work on MIDP20 device with `GLLibConfig.sound_useJSR135=true`

**Returns:**

- duration of the sound

### static void Snd_ForceExecOnThreadOnGamePause ( ) [static, package, inherited]

When the game receive an interrupt, it can be important to force execution of the sound thread.

Its going to work only if `GLLib.s_game_isPaused` is True.

### static final void Snd_FreeChannel ( int channel ) [static, package, inherited]

Free sound channel from the player asigned to it.
**Parameters:**

channel  Channel to prepare the sound on.

---

Object Snd_GetChannelPlayer ( int channel ) [package, inherited]

---

**static int Snd_GetChannelVolume ( int channel ) throws Exception [static, package, inherited]**

Get the current volume of a channel.

Scaled to (0-100) (API independent, Master Volume independent)

**Parameters:**

channel  whose current volume we want

**Returns:**

the volume with which this channel is playing (0-100)

**Note:**

Not implemented for Nokia API

---

**static int Snd_GetCurrentSoundIndex ( int nChannel ) [static, package, inherited]**

Get the index of the sound loaded on the channel.

**Parameters:**

nChannel  Channel to query for the loaded sound index.

**Returns:**

The sound index of the channel. or -1 if there is no sound loaded.

**Note:**

The dummy sound are never loaded and will always return -1.

---

**static void Snd_Init ( int nbSoundSlot ) throws Exception [static, package, inherited]**
Allocate sound player and resources.

Init sound container.

**Parameters:**

\[ nbSoundSlot \]

**Parameters:**

nbSoundSlot  number of sounds file you want to have in memory.

**Note:**

This call does not load the sound files (mid,wav,mp3...) in memory.

Also this is not the number of sound your phone can play at once (Channels).

---

static boolean Snd_IsPlaying ( int channel ) throws Exception [static, protected, inherited]

Return true if a sound is currently playing on channel.

**Parameters:**

channel  Channel where the sound is suposed to play.

**Note:**

dummy sound are always NOT playing.

---

static void Snd_LoadSound ( byte[] soundData, 
int nMIME, 
int index, 
boolean bCacheThisSound ) throws Exception [static, package]

Load a sound file/resource from data package.

**Parameters:**

soundData  Byte array containing the sound data to be loaded.

nMIME  MIME type of the data in soundData.

index  Index of the sound, to ask to play/stop/pause/free... it.

bCacheThisSound  this sound will be cached. Works ONLY if sound_useCachedPlayers is true.
Warning:
index Must be smaller than the Slot count when you initialized the player.

static void Snd_LoadSound (byte[] soundData, int nMIME, int index) throws Exception [static, package, inherited]

Load a sound file/resource from data package.

Parameters:
- soundData: Byte array containing the sound data to be loaded.
- nMIME: MIME type of the data in soundData.
- index: Index of the sound, to ask to play/stop/pause/free... it.

Warning:
index Must be smaller than the Slot count when you initialized the player.

static void Snd_LoadSound (String dataFileName, int resourceIndex, boolean bCacheThisSound) throws Exception [static, package, inherited]

Load a sound file/resource from data package.

Parameters:
- dataFileName: Path+filename of the file to load the sound from.
- resourceIndex: Resource index of sound to load in the dataFileName resource file.
- bCacheThisSound: this sound will be cached. Works ONLY if sound_useCachedPlayers is true.

Warning:
resourceIndex is the index in the resource pack BUT it will be also the index to play/stop the sound. So make sure that all of your sound are at the begining of a pack, because if your sound are at the end of a resource pack with resourceIndex
ranging from 100 to 110, you will have to init the sound library with 120 slot or you will have out of bound exception.

Note:
If resourceIndex is invalid no error is raised because some version of the game have less sound than others, doing this, we can keep the code unchanged for each version.

This function is using **GLLib** package management.

**See also:**
**GLLib.Pack_ReadData**

```java
static void Snd_LoadSound ( String dataFileName, int resourceIndex )
throws Exception [static, package, inherited]
```

Load a sound file/resource from data package.

**Parameters:**
- `dataFileName`  Path+filename of the file to load the sound from.
- `resourceIndex` Resource index of sound to load in the dataFileName resource file.

**Warning:**
`resourceIndex` is the index in the resource pack BUT it will be also the index to play/stop the sound. So make sure that all of your sound are at the begining of a pack, because if your sound are at the end of a resource pack with resourceIndex ranging from 100 to 110, you will have to init the sound library with 120 slot or you will have out of bound exception.

**Note:**
If resourceIndex is invalid no error is raised because some version of the game have less sound than others, doing this, we can keep the code unchanged for each version.

This function is using **GLLib** package management.

**See also:**
GLLib.Pack_ReadData

**static long Snd_MediaTimeGet ( int channel )** [static, package, inherited]

get current media time

**Parameters:**

channel - channel on which the midi sound is currently playing

**Returns:**
current media time or -1 if unknown or not recoverable

**static long Snd_MediaTimeSet ( int channel, long time )** [static, package, inherited]

set current media time

**Parameters:**

channel - channel on which the midi sound is currently playing
time - time to set media on

**Returns:**
current media time or -1 if unknown or not recoverable or failed

**static void Snd_MidiPlayNote ( int channel, int MIDICchannel, int note, int volume )** throws Exception [static, package, inherited]

play a note on a midi channel.
a clean implementation should issue a note off( volume=0) event to finish playing a note

**Parameters:**
adjust volume of a midi channel

**Parameters:**

- **channel** - channel on which the midi sound is currently playing
- **MIDIChannel** - midi channel to alter [0-15]
- **volume** - volume to set [0=off, 126]

**Returns:**

true if it was able to set volume

**Note:**

will only work on MIDP20 device with

`GLLibConfig.sound_useJSR135=true`

this command may be overridde by the midi file itself if it issues a setvolume command itself later

**Pause sound on a channel.**

If this channel is not currently playing a sound, nothing will be done.

**Parameters:**

- **channel** - channel on which the sound is currently playing
```
static void Snd_Play ( int channel,
   int index,
   int loop,
   int volume,
   int priority
)  
```

Play a sound on a channel.

If the sound was not prepared it will be done before playing.

**Parameters:**
- `channel` Channel to play the sound on.
- `index` Index of BGM to play.
- `loop` Number of time to play this sound. (0:infinite)
- `volume` Volume of BGM (0-100).
- `priority` Priority of this sound. (0:biggest - 15:lowest)

**See also:**
- Snd_prepareSound

```
static void Snd_PrepareSound ( int channel,
   int index,
   int priority
)  
```

Prepare a sound to be played on a channel.

If there is already a sound prepared, this call will be ignored if the new priority is lower than the current priority. If the new priority is higher of equal, this new request will be executed.

**Parameters:**
- `channel` Channel to prepare the sound on.
- `index` Index of the sound to be played.
- `priority` Priority of this request. (0:highest - 15:lowest)
**static void Snd Quit ( ) throws Exception [static, package, inherited]**

Deallocate all sound ressources and players.

**Note:**

All sound loaded with Snd_loadSound are going to be freed from memory.

**static int Snd_RateGet ( int channel ) [static, package, inherited]**

get rate of a sound

**Parameters:**

* channel - channel of the sound to alter the rate

**Note:**

will only work on MIDP20 device with 

```java
GLLibConfig.sound_useJSR135=true
```

**Returns:**

current sound rate (-1 if unable to get it)

**static int Snd_RateGetMax ( int channel ) throws Exception [static, package, inherited]**

get max rate of a sound

**Parameters:**

* channel - channel of the sound

**Note:**

will only work on MIDP20 device with 

```java
GLLibConfig.sound_useJSR135=true
```

**Returns:**

max rate value
static int Snd_RateGetMin ( int  \textit{channel}  ) throws Exception \texttt{[static, package, inherited]} \par

get min rate of a sound \par

\textbf{Parameters:} \par
\hspace{1em} \textit{channel} - channel of the sound \par

\textbf{Note:} \par
will only work on MIDP20 device with \par
\hspace{1em} \texttt{GLLibConfig.sound\_useJSR135=\text{true}} \par

\textbf{Returns:} \par
min rate value \par

---

static boolean Snd_RateSet ( int  \textit{channel}, \par
\hspace{1em} int  \textit{rate}  ) throws Exception \texttt{[static, package, inherited]} \par

set rate of a sound \par

\textbf{Parameters:} \par
\hspace{1em} \textit{channel} - channel of the sound to alter the rate \par
\hspace{1em} \textit{rate} - rate to set (100000 = 100 % speed) \par

\textbf{Note:} \par
will only work on MIDP20 device with \par
\hspace{1em} \texttt{GLLibConfig.sound\_useJSR135=\text{true}} \par

\textbf{Returns:} \par
true if it was able to set rate \par

---

static final void Snd_Resume ( int  \textit{channel}  ) \texttt{[static, package, inherited]} \par

Resume currently paused sound on a channel. \par
If there is no paused sound, nothing will be done.
Parameters:

channel Channel of the sound to resume.

\[
\text{static void Snd_SetMasterVolume ( int } \text{ volume } \text{ ) throws Exception [static, package, inherited]}
\]

Change master volume value.

**Parameters:**

- **volume** The new volume value. (0 - 100)

**Note:**

It may not be used until the next Snd_play.

\[
\text{static void Snd_SetMediaDuration ( int } \text{ index, }
\]

\[
\text{ int } \text{ duration } \text{ ) throws Exception [static, package, inherited]}
\]

Sets duration for the sound (Use with \text{GLLibConfig.sound_useFakeMediaDuration} == true).

**Parameters:**

- **index** Sound slot index
- **duration** duration.

**Note:**

Use with \text{GLLibConfig.sound_useFakeMediaDuration} == true

\[
\text{static final void Snd_Stop ( int } \text{ channel } \text{ ) [static, package, inherited]}
\]

Stop sound on a channel.

If this channel is not currently playing a sound, nothing will be done.

**Parameters:**

- **channel** Channel of the sound to stop.
### static void Snd_StopAllSounds () [static, package, inherited]

Stop all sounds on all channel.

### static int Snd_TempoGet ( int channel ) throws Exception [static, package, inherited]

get tempo of a sound

**Parameters:**

- *channel* - channel of the sound to get the tempo

**Note:**

will only work on MIDP20 device with

```java
GLLibConfig.sound_useJSR135=true
```

**Returns:**

tempo or -1 if unable to get it

### static boolean Snd_TempoSet ( int channel, int tempo ) throws Exception [static, package, inherited]

set tempo of a sound

**Parameters:**

- *channel* - channel of the sound to alter the tempo
- *tempo* - tempo to set (120000 = 120 beats per minute)

**Note:**

will only work on MIDP20 device with

```java
GLLibConfig.sound_useJSR135=true
```

it is usually better to use Snd_SetRate instead of altering the tempo of the sound

**Returns:**

true if it was able to set tempo
static void Snd_UnLoadSound ( int index ) throws Exception

Unload a sound resource from memory.

No exception will thrown if the sound is already unloaded or if the index is invalid.

Parameters:
index  Index of sound to unload.

static void Snd_Update ( )

Update sound engine status all sound ressource.

This call HAS to be called at every game loop.

Note:
If your device is using the Threaded system, this call will do nothing. It is a good idea to keep calling it any way to prevent errors/bugs with unthreaded devices.
Variable Documentation

```java
final int k_snd_priority_highest = 0 [static, package, inherited]

Sound priorities constant.
```

```java
final int k_snd_priority_lowest = 15 [static, package, inherited]

Sound priorities constant.
```

```java
final int k_snd_priority_normal = 7 [static, package, inherited]

Sound priorities constant.
```

```java
String [] s_data_mimeType [static, package, inherited]

Sound Mime Container, to know the sound type of each slots.
```

```java
int s_snd_masterVolume [static, package, inherited]

Sound engine master volume (0-100). The volume will be scaled to the phone range.
```

```java
int s_snd_maxNbSoundSlot [static, package, inherited]

Sound maximum slot count for this game. The maximum number of sounds that can be loaded into memory.
```
GLLibPlayer : Tileset

[GLLibPlayer]

tile set loading/drawing function More...
## Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static void</strong></td>
<td><strong>Tileset_Destroy</strong> (int nLayer, boolean bFreeBufferImage) Delete a layer from the player.</td>
</tr>
<tr>
<td><strong>static void</strong></td>
<td><strong>Tileset_Destroy</strong> (int nLayer) Delete a layer from the player.</td>
</tr>
<tr>
<td><strong>static void</strong></td>
<td><strong>Tileset_Draw</strong> (Graphics g, int dx, int dy, int nLayer) Draw a specific onto destination Graphics.</td>
</tr>
<tr>
<td><strong>static void</strong></td>
<td><strong>Tileset_Draw</strong> (Graphics g, int nLayer) Draw a specific onto destination Graphics.</td>
</tr>
<tr>
<td><strong>static final Graphics</strong></td>
<td><strong>Tileset_GetBufferGraphics</strong> (int p_iLayer, int p_iImage) Get the graphics context of one of the images being used as a buffer for this layer.</td>
</tr>
<tr>
<td><strong>static final Graphics</strong></td>
<td><strong>Tileset_GetBufferGraphics</strong> (int p_iLayer) Get the graphics context of the image being used as a buffer for this layer.</td>
</tr>
<tr>
<td><strong>static final Image</strong></td>
<td><strong>Tileset_GetBufferImage</strong> (int p_iLayer, int p_image) Get one of the images that is being used as a buffer for this layer.</td>
</tr>
<tr>
<td><strong>static final Image</strong></td>
<td><strong>Tileset_GetBufferImage</strong> (int p_iLayer) Get the image that is the buffer for this layer.</td>
</tr>
<tr>
<td><strong>static final int[]</strong></td>
<td><strong>Tileset_GetCamera</strong> (int nLayer) Get the camera position of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetCameraX</strong> (int nLayer) Get the camera X position of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetCameraY</strong> (int nLayer) Get the camera Y position of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetLayerHeight</strong> (int nLayer) Get the pixel height of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetLayerTileCountHeight</strong> (int nLayer) Get the tile count height of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetLayerTileCountWidth</strong> (int nLayer) Get the tile count width of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetLayerWidth</strong> (int nLayer) Get the pixel width of a specific layer.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetTile</strong> (int nLayer, int x, int y) Get value of a specific tile.</td>
</tr>
<tr>
<td><strong>static final int</strong></td>
<td><strong>Tileset_GetTileFlags</strong> (int nLayer, int x, int y) Get the flags information of a specific tile.</td>
</tr>
<tr>
<td><strong>static void</strong></td>
<td><strong>Tileset_Init</strong> (int nDestWidth, int nDestHeight, int nTileWidth, int nTileHeight)</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>Tileset_LoadLayer</code></td>
<td>(int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, int iUseCB, int origin, int wrappingX, int wrappingY)</td>
</tr>
<tr>
<td></td>
<td>Load a tileset layer into the player.</td>
</tr>
<tr>
<td><code>Tileset_LoadLayer</code></td>
<td>(int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, boolean bUseCB, int origin, int wrappingX, int wrappingY)</td>
</tr>
<tr>
<td></td>
<td>Load a tileset layer into the player.</td>
</tr>
<tr>
<td><code>Tileset_SetCamera</code></td>
<td>(int nLayer, int x, int y)</td>
</tr>
<tr>
<td></td>
<td>Set the camera position of a specific layer.</td>
</tr>
<tr>
<td><code>Tileset_Update</code></td>
<td>(int nLayer)</td>
</tr>
<tr>
<td></td>
<td>Update a specific layer circular buffer (back buffer) but does nothing if this layer is not using a circular buffer.</td>
</tr>
</tbody>
</table>
## Variables

<table>
<thead>
<tr>
<th>static final int</th>
<th>WRAP_CLAMP = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>wrappng parameter, specify that the tileset is to be repeated only once</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final int</th>
<th>WRAP_REPEAT = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>wrappng parameter, specify that the tileset is to be repeated indefinitely</td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

tile set loading/drawing function

GLLibPlayer does not need to be instanciated to use those function
Function Documentation

```java
static void Tileset_Destroy ( int nLayer,
    boolean bFreeBufferImage
) [static, package, inherited]

Delete a layer from the player.

Parameters:
    nLayer  Layer to delete.
    bFreeBufferImage  Free buffer image
```

```java
static void Tileset_Destroy ( int nLayer ) [static, package, inherited]

Delete a layer from the player.

Image and data are going to be freed.

Parameters:
    nLayer  Layer to delete.
```

```java
static void Tileset_Draw ( Graphics g,
    int dx,
    int dy,
    int nLayer
) [static, package, inherited]

Draw a specific onto destination Graphics.

If this layer is using a circular buffer it will be drawn to destination using correct order. If this layer is not using a circular buffer, every tile will be drawn to fill the destination.

Parameters:
    g  Graphics destination.
    dx  X offset in the destination.
    dy  Y offset in the destination.
```
Draw a specific onto destination Graphics.

If this layer is using a circular buffer it will be drawn to destination using correct order. If this layer is not using a circular buffer, every tile will be drawn to fill the destination.

**Parameters:**

- `g`: Graphics destination.
- `nLayer`: Layer to draw. Use -1 to draw every valid layer.

**Returns:**

the graphics context of the buffer
layer.

**Parameters:**

  * p_iLayer  Layer whose buffer graphics context we want

**Returns:**

  the graphics context of the buffer

```java
static final Image Tileset_GetBufferImage ( int p_iLayer,
                                            int p_iImage
                                      )  [static, package, inherited]
```

Get one of the images that is being used as a buffer for this layer.

**NOTE:** Current implementation always uses 1 image, so for now you can always use this method with 1 parameter

**Parameters:**

  * p_iLayer  Layer whose buffer we want as an image
  * p_iImage  Image index we want

**Returns:**

  the Image that is being used to buffer this layer

```java
static final Image Tileset_GetBufferImage ( int p_iLayer )  [static, package, inherited]
```

Get the image that is the buffer for this layer.

Defaults to first image (only one usually)

**Parameters:**

  * p_iLayer  Layer whose buffer we want as an image

**Returns:**

  the Image that is being used to buffer this layer

```java
static final int [] Tileset_GetCamera ( int nLayer )  [static, package, inherited]
```
Get the camera position of a specific layer.

**Parameters:**

$nLayer$  Layer of which you want to know the Camera position.

**Returns:**

a 2 int array with the x and y in pixel of the camera. (null if entry params are invalid).

**See also:**

Tileset_GetCameraX, Tileset_GetCameraY

**Deprecated:**

use Tileset_GetCameraX and Tileset_GetCameraY instead

```java
static final int Tileset_GetCameraX ( int nLayer ) [static, package, inherited]

Get the camera X position of a specific layer.

**Parameters:**

$nLayer$  Layer of which you want to know the Camera position.

**Returns:**

camera X position
```

```java
static final int Tileset_GetCameraY ( int nLayer ) [static, package, inherited]

Get the camera Y position of a specific layer.

**Parameters:**

$nLayer$  Layer of which you want to know the Camera position.

**Returns:**

camera Y position
```
### `static final int Tileset_GetLayerHeight ( int nLayer )` [static, package, inherited]

Get the pixel height of a specific layer.

**Parameters:**

- `nLayer`  Layer of the tile to query.

**Returns:**

the pixel height of the requested layer. (-1 if entry params are invalid).

### `static final int Tileset_GetLayerTileCountHeight ( int nLayer )` [static, package, inherited]

Get the tile count height of a specific layer.

**Parameters:**

- `nLayer`  Layer of the tile to query.

**Returns:**

the tile count height of the requested layer. (-1 if entry params are invalid).

### `static final int Tileset_GetLayerTileCountWidth ( int nLayer )` [static, package, inherited]

Get the tile count width of a specific layer.

**Parameters:**

- `nLayer`  Layer of the tile to query.

**Returns:**

the tile count width of the requested layer. (-1 if entry params are invalid).

### `static final int Tileset_GetLayerWidth ( int nLayer )` [static, package, inherited]

Get the pixel width of a specific layer.
Parameters:

\( nLayer \)  Layer of the tile to query.

Returns:

the pixel width of the requested layer. (-1 if entry params are invalid).

```
static final int Tileset_GetTile ( int nLayer, 
    int x, 
    int y 
)  [static, package, inherited]
```

Get value of a specific tile.

Parameters:

\( nLayer \)  Layer of the tile to query. 
\( x \) X Position of the tile to query. 
\( y \) Y Position of the tile to query.

Returns:

the value of the requested tile. (-1 if entry params are invalid).

```
static final int Tileset_GetTileFlags ( int nLayer, 
    int x, 
    int y 
)   [static, package, inherited]
```

Get the flags information of a specific tile.

Parameters:

\( nLayer \)  Layer of the tile to query. 
\( x \) X Position of the tile to query. 
\( y \) Y Position of the tile to query.

Returns:

the flags value of the requested tile. (-1 if entry params are invalid).
Initialize the GLLibPlayerTileSet engine.

You cannot use tileset before you call this function.

Parameters:

- `nDestWidth` Destination Width in pixel; Usually Screen Width.
- `nDestHeight` Destination Height in pixel; Usually Screen Height.
- `nTileWidth` tile width
- `nTileHeight` tile height

Load a tileset layer into the player.

Parameters:

- `nLayer` Layer to load. It will replace the information if this layer is already loaded.
- `MapSizes` Array of short containing this layer size in tile count. Array generated by Aurora (.game).
- `MapData` Array of byte containing the value of each position of the map. The Size of this array should be w*h in tile count. Byte array generated by Aurora (.game).
- `MapFlip` Array of byte containing the flip value of each tile on the map. The Size of this array should be w*h in tile count. Byte array generated by Aurora (.game). can be null, in which case no flip will occurs on the tile
- `MapSprite` Sprite to use to draw each tile. If this sprite contains any frame, the engine will use PaintFrame to draw each tile of this layer. If the sprite contains only
module, the tile will be drawn with PaintModule.

possible values are

-1 to disable circular buffer

nLayer to allocate a circular buffer for this layer

0..n to reuse the previously allocated buffer of layer 0..n.

set the position of coord 0,0 in layer. 0.top left, 1.bottom left

set the position of coord 0,0 in tileset (either GLLib.TOP or GLLib.BOTTOM)

<table>
<thead>
<tr>
<th>iUseCB</th>
<th>possible values are</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>disable circular buffer</td>
</tr>
<tr>
<td>nLayer</td>
<td>allocate a circular buffer for this layer</td>
</tr>
<tr>
<td>0..n</td>
<td>reuse the previously allocated buffer of layer 0..n.</td>
</tr>
</tbody>
</table>

| origin | set the position of coord 0,0 in layer. 0.top left, 1.bottom left |
| origin | set the position of coord 0,0 in tileset (either GLLib.TOP or GLLib.BOTTOM) |

| wrappingX | tileset wrapping policy in X direction |
| wrappingY | tileset wrapping policy in Y direction |

static void Tileset_LoadLayer ( int nLayer,
byte[] MapSizes,
byte[] MapData,
byte[] MapFlip,
ASprite MapSprite,
boolean bUseCB,
int origin,
int wrappingX,
int wrappingY )

[static, package, inherited]

Load a tileset layer into the player.

Parameters:

- **nLayer** - Layer to load. It will replace the information if this layer is already loaded.
- **MapSizes** - Array of short containing this layer size in tile count. Array generated by Aurora (.game).
- **MapData** - Array of byte containing the value of each position of the map. The Size of this array should be w*h in tile count. Byte array generated by Aurora (.game).
- **MapFlip** - Array of byte containing the flip value of each tile on the map. The Size of this array should be w*h in tile count. Byte array generated by Aurora (.game). can be null, in which case no flip will occur on the tile
- **MapSprite** - Sprite to use to draw each tile. If this sprite contains any frame, the engine will use PaintFrame to draw each tile of this layer. If the sprite contains only module, the tile will be drawn with PaintModule.
- **bUseCB** - True if you want to use a Circular buffer (backbuffer) for this layer. Usually this should only be used with the background layer, when drawing this layer, it will overwrite what was on the destination, without transparency.

Note:

Using this will create an image of the size of the destination.
<table>
<thead>
<tr>
<th><strong>Parameters:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>origin</strong></td>
</tr>
<tr>
<td><strong>wrappingX</strong></td>
</tr>
<tr>
<td><strong>wrappingY</strong></td>
</tr>
<tr>
<td><strong>origin</strong></td>
</tr>
</tbody>
</table>

**Deprecated:** use `Tileset_LoadLayer(int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, int iUseCB, int origin, int wrappingX, int wrappingY)` instead

```java
static final void Tileset_SetCamera ( int nLayer, int x, int y ) [static, package, inherited]
```

Set the camera position of a specific layer.

**Parameters:**
- **nLayer** Layer of which you want to set the Camera position.
- **x** the new X Position of the camera.
- **y** the new Y Position of the camera.

```java
static void Tileset_Update ( int nLayer ) [static, package, inherited]
```

Update a specific layer circular buffer (back buffer) but does nothing if this layer is not using a circular buffer.

**Parameters:**
- **nLayer** Layer to update. Use -1 to update every valid layer.

**Deprecated:**
do not use anymore
Variable Documentation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>final int WRAP_CLAMP = 0 [static, inherited]</td>
<td>wrappng parameter, specify that the tileset is to be repeated only once</td>
</tr>
<tr>
<td>final int WRAP_REPEAT = 1 [static, inherited]</td>
<td>wrappng parameter, specify that the tileset is to be repeated indefinitely</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
XPlayer handles communication with the Gameloft server More...
## Modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP (used by XPlayer and License)</td>
<td>HTTP connection class.</td>
</tr>
<tr>
<td>TCP (used by XPlayer)</td>
<td>TCP connection class.</td>
</tr>
<tr>
<td>XPlayer core</td>
<td>The base class for XPlayer communication with the Gameloft server.</td>
</tr>
<tr>
<td>Xplayer Error Code</td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

handles communication with the Gameloft server
HTTP (used by XPlayer and License)
[XPlayer]

HTTP connection class. More...
### Classes

<table>
<thead>
<tr>
<th>class</th>
<th>HTTP</th>
</tr>
</thead>
</table>

Detailed Description

HTTP connection class.

Provides services for creating and managing HTTP requests. This is non-blocking and runs as a separate thread on the device. It also provides a method for canceling an ongoing transaction.

How to use:

1. create an instance
2. call sendByGet() to begin a transaction
3. call isInProgress() to find out the status of an on going transaction
4. call cancel() to stop at anytime
5. check m_response for return from server
6. call cleanup() to free resources (will free the m_response buffer, so use it before cleanup)

Author:
Gameloft Online Team

Version:
1.0.9

Copyright:
Gameloft SA 2006
TCP (used by XPlayer)

[XPlayer]

TCP connection class. More...
Classes

class TCP
Detailed Description

**TCP** connection class.

Provides services for creating and managing **TCP** connections to the multiplayer server.

How to use:

1. Create an instance
2. Call `connect()` to establish the connection.
3. Send the establish connection packages, one on the receive connection and one on the sending connection.
4. Call `sendPacket()` to send a message to the server.
5. Call `recvPacket()` to check for server messages.
6. Call `disconnect()` to disconnect from the multiplayer server and cleanup resources.

**Author:**
Gamloft Online Team

**Version:**
1.0.9

**Copyright:**
Gameloft SA © 2006
XPlayer core

[XPlayer]

The base class for XPlayer communication with the Gameloft server.

More...
# Classes

<table>
<thead>
<tr>
<th>class</th>
<th>XPlayer</th>
</tr>
</thead>
</table>

Detailed Description

The base class for XPlayer communication with the Gameloft server.

Important:
Make sure the API connects to the correct URL and gameserver!!! Ask interactivity team for the correct urls.

Control the set of features offered by the api using the Configuration class.

The api consists of two parts: XPlayer Highscores and XPlayer Multiplayer.

XPlayer Highscores

Provides services for registering a nickname, logging in, sending highscores, viewing leaderboard(s), rating and recommending games, viewing stats.

The main mechanism for passing requests to the Gameloft server is by using HTTP transactions. Every action that results in contacting the server has a pair of asynchronous functions, a send function and a handle.

The send function builds up the HTTP message with the request and initiates the transaction (sends that message to the server).

The handle manages the connection and any events that might occur, for example a response from the server, a timeout or an error. It sets an error code indicating the status of the current connection which can be viewed by the means of the getLastError() function provided.

This means that the usual cycle of a request is the following:

1. Initiate the transaction by sending the request.
2. While the result from the server is pending and a timeout has not occurred, call the handle and check the error code returned with getLastError();
3. Analysing the error code returned, determine a correct response (if the result is still pending, wait some more, if timed out or some other error display a meaningful message, etc.)

Following is a short example of a generic game rate sending cycle:

Common practice is to use a 4-step request process.

- First we have a SET step which initiates the request.
- Second is a WAIT step for response waiting.
- Third is the PROCESS step which determines the outcome of the request. It can either end successfully or return some error.
- Last is the DISPLAY step in which the results are displayed.

```java
switch (_online_substate)
{
    case OnlineSubstate.SET:
    {
        The user's rate is sent to the server.
        sendRateGame(rate);
        We go to the WAIT step of the request.
        setOnlineSubstate(OnlineSubstate.WAIT);
        break;
    }

    case OnlineSubstate.WAIT:
    {
        ...// Here there should be a test on a keypress event which if true would call the cancel() method of the XPlayer class, canceling any request. This would provide the user with an option to cancel the transaction.

        We call the handle
        handleRateGame();
        and get the error code returned by it.
        _errCode = getLastError();

        If the connection is not ongoing (it has finished, either successfully or abnormally), we move on to PROCESS. Otherwise the result is still pending we keep waiting.
        if (_errCode != XPlayer.Error.ERROR_PENDING)
        {
```
cleanup();
setOnlineSubstate(OnlineSubstate.PROCESS);
}
break;
}

The request finished somehow so we must determine the way it did. We'll therefore check the error code returned.
case OnlineSubstate.PROCESS:
{
    switch (_errCode)
    {
        case XPlayer.Error.ERROR_NONE:
        {
            // No error - do anything that might be needed here and then go to displaying the result.
            setOnlineSubstate(OnlineSubstate.DISPLAY);
            break;
        }
        case XPlayer.Error.ERROR_CONNECTION:
        {
            // A connection error occurred - treat the error here, inform the user of its appearance and then go to a new state. We should also provide the user the means to retry the request.
            break;
        }
        default:
        {
            // Some other error occurred. Also, an error message should be displayed and an option to retry provided.
            break;
        }
    }
break;
}

SCORE SENDING FUNCTIONS:
Depending on the game, different score sending functions can be used.
• If the game has supplemental data attached to each leaderboard, use the supplemental data functions. Posting a score to a leaderboard that has supplemental data will FAIL if any of the supplemental data fields are missing from the message!! ALL supplemental data for a given level MUST be present in the post score message for this level or it will fail. So for leaderboards with supplemental data use the sendHighscoreWithSupplementalData() function.

• If the game has multiple leaderboards, all the scores can be sent using a single http request, with or without supplemental data. When multiple scores are to be sent, the procedure is: reset the multiple score buffer, add the scores in the buffer and call send.

• If the game only has one leaderboard, don't send the level parameter! Call this functions with level = -1.

**XPlayer Multiplayer**

Provides services for creating a multiplayer game, joining a game, listing the available games and playing a multiplayer game.

The communication with the server is performed using **TCP** streams. The communication protocol is based on requests, similar to the **HTTP** requests from the Highscore XPlayer. There are some messages though that arrive to the client without a previous request being made.

The messages exchanged between the server and the client are:

• Connection messages - used to establish and terminate the physical connection.
• Game messages - requests, push messages and in game messages.

The connection messages and the game request messages work in a similar way to the requests in the Highscore XPlayer, with send and handle functions corresponding to each function. These functions are to be used the same way as the **HTTP** requests from XPlayer Highscore, even though the communication lies on **TCP** in this case. Look at the example above for sample use.
The push messages are sent by the server to inform the client that something has happened on the server side: some player just joined the current game, some player left, the game has started or somebody finished the game. To make sure the push messages are processed, the game must call `mpHandleUpdates()` on every cycle while being in a state where push messages are expected to be received (in session waiting start, in game playing).

The in game messages are sent by the directly by the caller class and they are forwarded by the server to all the other clients that are engaged in a multiplayer game with the current player. The api does not do any processing of those messages. To make sure the in game messages are received and stored in the in game message queue, the game must call `mpHandleGameData()` on every cycle while the multiplayer game is in progress.

Here is how it works:

Before performing any multiplayer actions, the player must establish a connection and login on the multiplayer server. This is achieved using the `mpSendEstablishConnection()/mpHandleEstablishConnection()` and `mpSendLogin()/mpHandleLogin()` functions. According to the design of the game, the connection establish/login operations can be performed when a multiplayer menu is accesses, with disconnection occuring when the menu is left, or they can be performed before any multiplayer game, with disconnections occuring when the game is finished.

After connection and login, the player can choose to go to quick game, create a new game or list the available games. The quick game option returns the first available game on the server if any, create game makes a new game on the multiplayer server and list games returns a list of all the games in the server that can be joined.

If a player creates a game, he is the master of the game and can kick out a player that joined or start the game when enough players have joined. If the master leaves the session, the game is terminated and everybody else in the game is kicked out.

A game can be joined using the name of the game. The name can be
chosen from the sessions list.

After enough players (2 usually) are in a game, the master can choose to start the game. The other player will receive the start response as a push message.

After the game is started, the players can send game specific messages to each other.

The game is finished when everybody finishes the game and sends a finish message, or when some players have left and the game cannot go on.

Author:
  Gameloft Interactivity Team

Version:
  1.0.9

Copyright:
  Gameloft SA © 2005
Xplayer Error Code
[XPlayer]
Classes

interface XPlayer.Error

Generated on Tue Sep 23 23:05:31 2008 for GLib by doxygen 1.5.2
Here are the classes, structs, unions and interfaces with brief descriptions:

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASprite</td>
<td><em>ASprite</em> Aurora Sprite class</td>
</tr>
<tr>
<td>Canvas</td>
<td></td>
</tr>
<tr>
<td>GLKey</td>
<td>Game key code interface</td>
</tr>
<tr>
<td>GLLang</td>
<td>Language code</td>
</tr>
<tr>
<td>GLLib</td>
<td>Main class to do a game creation at Gameloft</td>
</tr>
<tr>
<td>GLLibConfig</td>
<td><em>GLLib</em> setup class</td>
</tr>
<tr>
<td>GLLibMidiSpectrumAnalyzer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Author: Nikolay Aleksiev</td>
</tr>
<tr>
<td>GLLibPathFinding</td>
<td><em>A</em> (A star) Pathfinding class</td>
</tr>
<tr>
<td>GLLibPlayer</td>
<td>Generic <em>GLLib</em> Player</td>
</tr>
<tr>
<td>GRPH</td>
<td><em>ATileSet</em> Aurora TileSet class</td>
</tr>
<tr>
<td>HTTP</td>
<td></td>
</tr>
<tr>
<td>License</td>
<td>The base class for <em>License</em> communication with the Gameloft server</td>
</tr>
<tr>
<td>License.Error</td>
<td><em>License</em> error codes</td>
</tr>
<tr>
<td>MessageType</td>
<td></td>
</tr>
<tr>
<td>Runnable</td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>XPlayer</td>
<td></td>
</tr>
<tr>
<td>XPlayer.ConnectLevel</td>
<td></td>
</tr>
<tr>
<td>XPlayer.Error</td>
<td></td>
</tr>
<tr>
<td>XPlayer.MessageType</td>
<td></td>
</tr>
</tbody>
</table>
ASprite Class Reference

[ASprite]

ASprite Aurora Sprite class. More...

List of all members.
Public Member Functions

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>ResizeCoords</code> ()</td>
<td>To be documented...</td>
</tr>
<tr>
<td>int</td>
<td><code>scaleX</code> (int x)</td>
<td>To be documented.</td>
</tr>
<tr>
<td>int</td>
<td><code>scaleY</code> (int y)</td>
<td>To be documented.</td>
</tr>
<tr>
<td>void</td>
<td><code>SetPool</code> (int poolIndex)</td>
<td>set cache pool of <code>ASprite</code></td>
</tr>
<tr>
<td>void</td>
<td><code>SetResizeParameters</code> (int spriteId, int resizeMode, boolean correctY)</td>
<td>To be documented.</td>
</tr>
</tbody>
</table>
## Static Public Member Functions

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td><strong>Crc32</strong></td>
<td>(byte buffer[], int start, int count, int crc)</td>
</tr>
<tr>
<td>static void</td>
<td><strong>InitCachePool</strong></td>
<td>(int poolCount)</td>
</tr>
<tr>
<td></td>
<td>Initialize Cache Pool</td>
<td></td>
</tr>
<tr>
<td>static void</td>
<td><strong>InitPoolSize</strong></td>
<td>(int poolIndex, int size)</td>
</tr>
<tr>
<td></td>
<td>Initialize pool size</td>
<td></td>
</tr>
<tr>
<td>static void</td>
<td><strong>ResetCachePool</strong></td>
<td>(int poolIndex)</td>
</tr>
<tr>
<td></td>
<td>Release pool</td>
<td></td>
</tr>
<tr>
<td>static void</td>
<td><strong>SetTempBuffer</strong></td>
<td>(Object pArray)</td>
</tr>
<tr>
<td></td>
<td>Sets the temporary buffer array to be used for decoding operations</td>
<td></td>
</tr>
</tbody>
</table>
### Static Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td>BASE</td>
<td>65521</td>
</tr>
<tr>
<td>static final int</td>
<td>BLOCK_INFO_SIZE</td>
<td>11</td>
</tr>
<tr>
<td>static final int</td>
<td>CRC32_POLYNOMIAL</td>
<td>0xEDB88320</td>
</tr>
<tr>
<td>static int</td>
<td>crcTable []</td>
<td>new int[256]</td>
</tr>
<tr>
<td>static final int</td>
<td>HEADER_LEVEL0_MAX_WBITS</td>
<td>30938</td>
</tr>
<tr>
<td>static final int</td>
<td>NMAX</td>
<td>5552</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_BLEND_BLACK</td>
<td>5</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_BLUE_CYAN</td>
<td>2</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_GREEN</td>
<td>3</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_GREY</td>
<td>4</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_INVISIBLE</td>
<td>0</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_ORIGINAL</td>
<td>-1</td>
</tr>
<tr>
<td>static final int</td>
<td>PAL_RED_YELLOW</td>
<td>1</td>
</tr>
<tr>
<td>static final int</td>
<td>PNG_INFO_SIZE</td>
<td>57</td>
</tr>
</tbody>
</table>
### Static Protected Attributes

<table>
<thead>
<tr>
<th>Static final byte</th>
<th>IDAT [] = { 'I', 'D', 'A', 'T' }</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final byte</td>
<td>IEND [] = { 'I', 'E', 'N', 'D' }</td>
</tr>
<tr>
<td>static final byte</td>
<td>IHDR [] = { 'I', 'H', 'D', 'R' }</td>
</tr>
<tr>
<td>static final byte</td>
<td>INFO32 [] = { 8, 6, 0, 0, 0 }</td>
</tr>
<tr>
<td>static final byte</td>
<td>INFO8 [] = { 8, 3, 0, 0, 0 }</td>
</tr>
<tr>
<td>static final byte</td>
<td>MAGIC [] = { -119, 80, 78, 71, 13, 10, 26, 10 }</td>
</tr>
<tr>
<td>static final byte</td>
<td>MAGIC_IEND = {0x00, 0x00, 0x00, 0x00, 0x49, 0x45, 0x4e, 0x44, (byte)0xae, 0x42, 0x60, (byte)0x82}</td>
</tr>
<tr>
<td>static final byte</td>
<td>PLTE [] = { 'P', 'L', 'T', 'E' }</td>
</tr>
<tr>
<td>static final byte</td>
<td>tRNS [] = { 't', 'R', 'N', 'S' }</td>
</tr>
</tbody>
</table>
### Package Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASprite</strong> ()</td>
<td>Empty constructor. Does nothing.</td>
</tr>
<tr>
<td>void <strong>BuildAnimCacheImages</strong> (int palette, int anim)</td>
<td>Create cache images for all modules all frame used by a animation.</td>
</tr>
<tr>
<td>void <strong>BuildCacheImages</strong> (int pal, int m1, int m2, int pal_copy)</td>
<td>Build Cache Images for this sprite.</td>
</tr>
<tr>
<td>void <strong>BuildFrameCacheImages</strong> (int palette, int frame)</td>
<td>Create cache images for all modules used by a frame.</td>
</tr>
<tr>
<td>int <strong>CountFrameModules</strong> (int frame)</td>
<td>Count the number of modules in a frame.</td>
</tr>
<tr>
<td>Object <strong>DecodeImage</strong> (int module)</td>
<td>Decodes loaded module data.</td>
</tr>
<tr>
<td>void <strong>DecodeImageToByteArray</strong> (byte[] dest, int module, boolean img2dRGBA, boolean half)</td>
<td>Decodes loaded module data to a byte array.</td>
</tr>
<tr>
<td>void <strong>DrawNumber</strong> (Graphics g, int num, int radix, int minDigit, int x, int y, int anchor, boolean restorecol)</td>
<td>Draws text given a number.</td>
</tr>
<tr>
<td>void <strong>DrawNumber</strong> (Graphics g, int num, int minDigit, int x, int y, int anchor)</td>
<td>Draws text given a number.</td>
</tr>
<tr>
<td>void <strong>DrawPage</strong> (Graphics g, String s, int x, int y, int anchor, int start, int end)</td>
<td>Draws a string with appropriate newline characters as a page given a start and end.</td>
</tr>
<tr>
<td>void <strong>DrawPage</strong> (Graphics g, String s, int x, int y, int anchor)</td>
<td>Draws a string with appropriate newline characters as a page.</td>
</tr>
<tr>
<td>void <strong>DrawPage</strong> (Graphics g, byte[] bs, int x, int y, int anchor)</td>
<td>Draws a byte array containing a string with appropriate newline characters as a page.</td>
</tr>
<tr>
<td>void <strong>DrawPageB</strong> (Graphics g, String s, short[] info, int x, int y, int startLine, int maxLines, int anchor)</td>
<td>Draws a string with appropriate newline characters and wrapping information as a page.</td>
</tr>
<tr>
<td>void <strong>DrawString</strong> (Graphics g, String s, int x, int y, int anchor, boolean restorecol)</td>
<td>Draws text given a string.</td>
</tr>
<tr>
<td>void <strong>DrawString</strong> (Graphics g, byte[] bs, int x, int y, int anchor, boolean restorecol)</td>
<td>Draws text given a byte array containing a string.</td>
</tr>
<tr>
<td>void <strong>DrawString</strong> (Graphics g, String s, int x, int y, int anchor)</td>
<td>Draws text given a string.</td>
</tr>
<tr>
<td>void <strong>DrawString</strong> (Graphics g, byte[] bs, int x, int y, int anchor)</td>
<td>Draws text given a string.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>void DrawStringOrChars (Graphics g, String s, char[] charBuff, int x, int y, int anchor, boolean restorecol)</td>
<td>Draws text given a byte array containing a string.</td>
</tr>
<tr>
<td>void FreeAnimCacheImages (int palette, int anim)</td>
<td>Free cache images for all modules of all frames used by a animation.</td>
</tr>
<tr>
<td>void FreeCacheData ()</td>
<td>Free all cached data.</td>
</tr>
<tr>
<td>void FreeFrameCacheImages (int palette, int frame)</td>
<td>Free cache images for all modules used by a frame.</td>
</tr>
<tr>
<td>void FreeMemory ()</td>
<td>Free data after the modules have been cached.</td>
</tr>
<tr>
<td>void FreeModuleImage (int nPal, int nMod)</td>
<td>Frees a cached module.</td>
</tr>
<tr>
<td>int GetAFrameFlags (int anim, int af.frame)</td>
<td>Gets the flags associated with a frame in an animation.</td>
</tr>
<tr>
<td>void GetAFrameRect (int anim, int af.frame, int rectIndex, int[] rect, int flags)</td>
<td>Get a FrameRect from a specific anim and af.frame.</td>
</tr>
<tr>
<td>void GetAFrameRect (int[] rc, int anim, int af.frame, int posX, int posY, int flags)</td>
<td>Gets the rectangle occupied by an animation frame if it were to be drawn at posX/posY with flags.</td>
</tr>
<tr>
<td>int GetAFrames (int anim)</td>
<td>Get the number of frames in an animation.</td>
</tr>
<tr>
<td>int GetAFramesOX (int v)</td>
<td>To be documented.</td>
</tr>
<tr>
<td>int GetAFramesOY (int v)</td>
<td>To be documented.</td>
</tr>
<tr>
<td>int GetAFrameTime (int anim, int af.frame)</td>
<td>Gets the time of a frame in an animation.</td>
</tr>
<tr>
<td>int GetAnimFrame (int anim, int af.frame)</td>
<td>Gets the ID of a frame inside an animation.</td>
</tr>
<tr>
<td>boolean GetBold ()</td>
<td>Gets a value that indicates whether this Font is Bold.</td>
</tr>
<tr>
<td>int GetCharFrame (int c)</td>
<td>Returns the current character map used by DrawString.</td>
</tr>
<tr>
<td>byte[] GetCharMap ()</td>
<td>new text rendering</td>
</tr>
<tr>
<td>short[][] GetCharMapShort ()</td>
<td>Gets the width of the given character using this sprite.</td>
</tr>
<tr>
<td>int GetCharSize (char c)</td>
<td>Gets the current character spacing for the sprite.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>int <code>GetCurrentMMapping()</code></td>
<td>Gets the current Module mapping index.</td>
</tr>
<tr>
<td>int <code>GetCurrentPalette()</code></td>
<td>Gets the current palette for the sprite @ returns the current palette.</td>
</tr>
<tr>
<td>int <code>GetFModuleOX(int v)</code></td>
<td>To be documented.</td>
</tr>
<tr>
<td>int <code>GetFModuleOY(int v)</code></td>
<td>To be documented.</td>
</tr>
<tr>
<td>void <code>GetFModuleRect(int[] rc, int frame, int fmodule, int posX, int posY, int flags, int hx, int hy)</code></td>
<td>Gets the rectangle occupied by a module in a frame if it were to be drawn at posX/posY with flags.</td>
</tr>
<tr>
<td>int <code>GetFModules(int frame)</code></td>
<td>Get the number of modules in a frame.</td>
</tr>
<tr>
<td>int <code>GetFontHeight()</code></td>
<td>Gets the font height of the sprite.</td>
</tr>
<tr>
<td>int <code>GetFrameCount()</code></td>
<td>Get the number of frames in the sprite.</td>
</tr>
<tr>
<td>int <code>GetFrameHeight(int frame)</code></td>
<td>Gets the height of a frame.</td>
</tr>
<tr>
<td>int[] <code>GetFrameMarkers(int frame)</code></td>
<td>Get all the marker module positions of a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModule(int frame, int fmodule)</code></td>
<td>Get the id of a module in a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModuleFlags(int frame, int fmodule)</code></td>
<td>Get the flags set on a module in a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModuleHeight(int frame, int fmodule)</code></td>
<td>Get the height of a module inside a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModulePalette(int frame, int fmodule)</code></td>
<td>Retrieve a frame module’s palette, if the sprite is exported using BS_FM_PALETTE flag.</td>
</tr>
<tr>
<td>int <code>GetFrameModuleWidth(int frame, int fmodule)</code></td>
<td>Get the width of a module inside a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModuleX(int frame, int fmodule)</code></td>
<td>Gets the X coordinate of a module inside a frame.</td>
</tr>
<tr>
<td>int <code>GetFrameModuleY(int frame, int fmodule)</code></td>
<td>Gets the Y coordinate of a module inside a frame.</td>
</tr>
<tr>
<td>void <code>GetFrameRect(int frame, int rectIndex, int[] rect, int flags)</code></td>
<td>Get a FrameRect from a specific frame.</td>
</tr>
<tr>
<td>void <code>GetFrameRect(int[] rc, int frame, int posX, int posY, int flags)</code></td>
<td>Gets the rectangle occupied by a frame if it were to be drawn at posX/posY with</td>
</tr>
</tbody>
</table>
void GetFrameRect (int rc, int frame, int posX, int posY, int flags, int hx, int hy)
Get the rectangle occupied by a frame if it were to be drawn at posX/posY with flags.

int GetFrameRectCount (int frame)
Get the FrameRect count for a specific frame.

int GetFrames ()
Get the number of frames in the sprite.

int GetFrameWidth (int frame)
Get the width of a frame.

int GetLineHeight ()
Get the current line height of the sprite.

int GetLineSpacing ()
Returns the current line spacing for the sprite.

int GetModuleCount ()
Get the number of modules in the sprite.

Object GetModuleData (int nModule, int nPalette)
Get the Module's Data of a specific Palette.

int GetModuleHeight (int module)
Get the height of a module.

int GetModuleHeightOrg (int module)
To be documented.

Image GetModuleImage (int nModule, int nPalette)
Get the Module's Image of a specific Palette.

void GetModuleRect (int[] rc, int module, int posX, int posY, int flags)
Get the rectangle occupied by a module if it were to be drawn at posX/posY with flags.

int GetModuleWidth (int module)
Get the width of a module.

int GetModuleWidthOrg (int module)
To be documented.

int GetModuleX (int module)
Get the x coordinate of a module.

int GetModuleY (int module)
Get the y coordinate of a module.

Object GetPalette (int nPalette)
Get a palette array.

int GetSpaceWidth ()
Get the current space width of the sprite.

boolean GetUnderline ()
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Load</strong></td>
<td>Loads the sprite from the given byte array and associated image.</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Loads the sprite from the given byte array and associated image.</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Loads the sprite from the given byte array with flags.</td>
</tr>
<tr>
<td><strong>Load</strong></td>
<td>Loads the sprite from the given byte array.</td>
</tr>
<tr>
<td><strong>ModifyPalette</strong></td>
<td>Modifies palette, original palette will turn into the alpha value, and set one and only color for all palette original palette = 0x00RRGGBB, color parameter = 0x00000000 result palette = 0x00000000.</td>
</tr>
<tr>
<td><strong>ModifyPaletteAlpha</strong></td>
<td>Modifies the alpha value of a palette.</td>
</tr>
<tr>
<td><strong>ModifyPaletteAlphaUsingAltPalette</strong></td>
<td>Modifies palette, original palette will have its alpha channel set using the second palette's RED channel.</td>
</tr>
<tr>
<td><strong>ModifyPaletteAlphaUsingLastPalette</strong></td>
<td>This is just a wrapper around ModifyPaletteAlphaUsingAltPalette which automatically uses the last palette for the alpha information.</td>
</tr>
<tr>
<td><strong>PaintAFrame</strong></td>
<td>Draws an animation frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintAFrame</strong></td>
<td>Draws an animation frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintFModule</strong></td>
<td>Draws a module from a frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintFModule</strong></td>
<td>Draws a module from a frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintFrame</strong></td>
<td>Draws a frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintFrame</strong></td>
<td>Draws a frame at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintModule</strong></td>
<td>Draws a module at PosX/PosY with Flags.</td>
</tr>
<tr>
<td><strong>PaintPrecomputedFrame</strong></td>
<td>Paints a pre-computed frame at x, y.</td>
</tr>
<tr>
<td><strong>PrecomputeAllFrames</strong></td>
<td>Precomputes all frames.</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>void PrecomputeFrame(Graphics g, int frame, int flags)</strong></td>
<td>Precompute a frame.</td>
</tr>
<tr>
<td><strong>void SetBold(boolean bBold)</strong></td>
<td>Sets the bold for the font.</td>
</tr>
<tr>
<td><strong>void SetCharMap(short[] pNewMap)</strong></td>
<td>Sets the character map used by DrawString for the new text rendering code.</td>
</tr>
<tr>
<td><strong>void SetCharMap(byte[] pNewMap)</strong></td>
<td>Sets the character map used by DrawString.</td>
</tr>
<tr>
<td><strong>void SetCharSpacing(int spacing)</strong></td>
<td>Sets the character spacing for the sprite.</td>
</tr>
<tr>
<td><strong>void SetCharSpacingToDefault()</strong></td>
<td>Reverts the character spacing to default for the sprite.</td>
</tr>
<tr>
<td><strong>void SetCurrentMMapping(int map)</strong></td>
<td>Sets the current Module mapping index.</td>
</tr>
<tr>
<td><strong>void SetCurrentPalette(int pal)</strong></td>
<td>Sets the current palette for the sprite.</td>
</tr>
<tr>
<td><strong>void SetDefaultFontMetrics()</strong></td>
<td>Sets the default values for font ascent, descent, height, space, and line spacing, based on the data found in the sprite file.</td>
</tr>
<tr>
<td><strong>void SetLineHeight(int nHeight)</strong></td>
<td>Sets the line height for the sprite.</td>
</tr>
<tr>
<td><strong>void SetLineHeightToDefault()</strong></td>
<td>Reverts the line height to the default for the sprite.</td>
</tr>
<tr>
<td><strong>void SetLineSpacing(int spacing)</strong></td>
<td>Sets the line spacing for the sprite.</td>
</tr>
<tr>
<td><strong>void SetLineSpacingToDefault()</strong></td>
<td>Reverts line spacing to the default for the sprite.</td>
</tr>
<tr>
<td><strong>void SetModuleImage(Image pImg, int nModule, int nPalette)</strong></td>
<td>Set the Module’s Image of a specific Palette.</td>
</tr>
<tr>
<td><strong>void SetModuleImagesArray(Object pData)</strong></td>
<td>Sets the module cached image array.</td>
</tr>
<tr>
<td><strong>void SetModuleMapping(int map, byte[] mmp)</strong></td>
<td>Set the module mapping at an index.</td>
</tr>
<tr>
<td><strong>void SetSpaceWidth(int spacing)</strong></td>
<td>Sets the space width of the sprite.</td>
</tr>
<tr>
<td><strong>void SetSpaceWidthToDefault()</strong></td>
<td>Reverts the space width to the default for the sprite.</td>
</tr>
<tr>
<td><strong>void SetUnderline(boolean bUnderline)</strong></td>
<td>Sets the underline for the font.</td>
</tr>
<tr>
<td><strong>String TextFitToFixedWidth(String str, int text_w)</strong></td>
<td>Fits the supplied string to a given width.</td>
</tr>
<tr>
<td>Type</td>
<td>Function Name</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>byte[]</td>
<td>TextFitToFixedWidth (byte[] str, int text_w)</td>
</tr>
<tr>
<td>int[]</td>
<td>TransformRGB (int[] image_data, int sizeX, int sizeY, int flags)</td>
</tr>
<tr>
<td>void</td>
<td>unload ()</td>
</tr>
<tr>
<td>void</td>
<td>UpdateNumberSize (int num, int radix, int minDigit)</td>
</tr>
<tr>
<td>void</td>
<td>UpdateStringOrCharsSize (String s, char[] charBuff)</td>
</tr>
<tr>
<td>void</td>
<td>UpdateStringSize (String s)</td>
</tr>
<tr>
<td>void</td>
<td>UpdateStringSize (byte[] bs)</td>
</tr>
<tr>
<td>short[]</td>
<td>WraptextB (String s, int width, int height)</td>
</tr>
</tbody>
</table>
## Static Package Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[static initializer]</td>
<td></td>
</tr>
<tr>
<td>static void DrawLine (Graphics g, int x1, int y1, int x2, int y2, int color, int radius, int initial)</td>
<td>Bresenham's Line-Drawing Algorithm.</td>
</tr>
<tr>
<td>static short[] GenPalette (int type, short[] pal)</td>
<td>Generates a palette based on another palette.</td>
</tr>
<tr>
<td>static int[] GenPalette (int type, int[] pal)</td>
<td>Generates a palette based on another palette.</td>
</tr>
<tr>
<td>static int GetChars (char[] charBuf, int i, int radix, int minDigit)</td>
<td></td>
</tr>
<tr>
<td>static int GetCurrentStringHeight ()</td>
<td>Gets the current string height used by DrawString.</td>
</tr>
<tr>
<td>static int GetCurrentStringWidth ()</td>
<td>Gets the current string width used by DrawString.</td>
</tr>
<tr>
<td>static void SetCharMapStatic (byte[] pNewMap)</td>
<td>Sets the character map used by DrawString from a static context.</td>
</tr>
<tr>
<td>static void SetGraphics (Graphics g1)</td>
<td>Sets the current graphics context.</td>
</tr>
<tr>
<td>static void SetSubString (int i1, int i2)</td>
<td>Set the current substring.</td>
</tr>
<tr>
<td>static int StringTokenize (String s, int index1, int index2, char token, int[] indices)</td>
<td>Tokenize a string given a separating token character.</td>
</tr>
</tbody>
</table>
### Package Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[]</td>
<td>_aframes</td>
</tr>
<tr>
<td>byte[]</td>
<td>_aframes_flags</td>
</tr>
<tr>
<td>byte[]</td>
<td>_aframes_frame</td>
</tr>
<tr>
<td>byte[]</td>
<td>_aframes_ox_byte</td>
</tr>
<tr>
<td>short[]</td>
<td>_aframes_ox_short</td>
</tr>
<tr>
<td>byte[]</td>
<td>_aframes_oy_byte</td>
</tr>
<tr>
<td>short[]</td>
<td>_aframes_oy_short</td>
</tr>
<tr>
<td>byte[]</td>
<td>_aframes_time</td>
</tr>
<tr>
<td>boolean</td>
<td>_alpha</td>
</tr>
<tr>
<td>short[]</td>
<td>_anims_af_start</td>
</tr>
<tr>
<td>byte[]</td>
<td>_anims_naf</td>
</tr>
<tr>
<td>int[][]</td>
<td>_aryPrecomputedFlags</td>
</tr>
<tr>
<td>Object[][]</td>
<td>_aryPrecomputedImages</td>
</tr>
<tr>
<td>short[][]</td>
<td>_aryPrecomputedImgX</td>
</tr>
<tr>
<td>short[][]</td>
<td>_aryPrecomputedImgY</td>
</tr>
<tr>
<td>short[][]</td>
<td>_aryPrecomputedSizeX</td>
</tr>
<tr>
<td>short[][]</td>
<td>_aryPrecomputedSizeY</td>
</tr>
<tr>
<td>short[]</td>
<td>_aryPrecomputedX</td>
</tr>
<tr>
<td>short[]</td>
<td>_aryPrecomputedY</td>
</tr>
<tr>
<td>int</td>
<td>_bs_flags</td>
</tr>
<tr>
<td>boolean</td>
<td>_bTraceNow</td>
</tr>
<tr>
<td>int</td>
<td>_colors</td>
</tr>
<tr>
<td>int</td>
<td>_cur_pal</td>
</tr>
<tr>
<td>int</td>
<td>_cur_pool = -1</td>
</tr>
<tr>
<td></td>
<td>which pool the cached images will be created in */</td>
</tr>
<tr>
<td>short[]</td>
<td>_data_format</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules_flags</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules_id</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules_ox_byte</td>
</tr>
<tr>
<td>short[]</td>
<td>_fmodules_ox_short</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules_oy_byte</td>
</tr>
<tr>
<td>short[]</td>
<td>_fmodules_oy_short</td>
</tr>
<tr>
<td>byte[]</td>
<td>_fmodules_pal</td>
</tr>
<tr>
<td>byte[]</td>
<td>_frames_col</td>
</tr>
<tr>
<td>short[]</td>
<td>_frames_col_short</td>
</tr>
<tr>
<td>short[]</td>
<td>_frames_fm_start</td>
</tr>
<tr>
<td>byte[]</td>
<td>_frames_nfm</td>
</tr>
</tbody>
</table>
byte[] _frames_rc
short[] _frames_rc_short
byte[] _frames_rects
short[] _frames_rects_short
short[] _frames_rects_start
byte[][] _gifHeader
int[] _header_size
int _i64rle_color_bits
int _i64rle_color_mask
Image[] _main_image
short[][] _map
byte[] _module_colors_byte
int[] _module_colors_int
Image[][] _module_image_imageAA
Image[][][] _module_image_imageAAA
int[][][] _module_image_intAAA
byte[] _module_types
byte[] _modules_data
int[] _modules_data_off_int
short[] _modules_data_off_short
short[] _modules_extra_info
short[] _modules_extra_pointer
byte[] _modules_h_byte
short[] _modules_h_scaled
short[] _modules_h_short
short[][][] _modules_image_shortAAA
byte[] _modules_usage
byte[] _modules_w_byte
short[] _modules_w_scaled
short[] _modules_w_short
byte[] _modules_x_byte
short[] _modules_x_short
byte[] _modules_y_byte
short[] _modules_y_short
int _nModules
int _old_pal
byte[] _pal_data
int[][] _pal_int
short[][] _pal_short
int[] _palettes
int[] _PNG_packed_IDAT_ADLER
int[] _PNG_packed_IDAT_CRC
int[] _PNG_packed_IHDR_CRC
<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>int[]</td>
<td>_PNG_packed_PLTE_CRC</td>
</tr>
<tr>
<td>int[]</td>
<td>_PNG_packed_tRNS_CRC</td>
</tr>
<tr>
<td>byte[][]</td>
<td>_transp</td>
</tr>
<tr>
<td>int[]</td>
<td>_w_pos</td>
</tr>
<tr>
<td>int</td>
<td>hRef</td>
</tr>
<tr>
<td>int</td>
<td>hTarget</td>
</tr>
<tr>
<td>boolean</td>
<td>mResizeCorrectY = false</td>
</tr>
<tr>
<td>int</td>
<td>wRef</td>
</tr>
<tr>
<td>int</td>
<td>wTarget</td>
</tr>
<tr>
<td>int</td>
<td>xRatio</td>
</tr>
<tr>
<td>int</td>
<td>yRatio</td>
</tr>
</tbody>
</table>
## Static Package Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static byte[]</td>
<td>_buffer_index</td>
<td></td>
</tr>
<tr>
<td>static Graphics</td>
<td>_graphics</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_images_count</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_images_size</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_index1 = -1</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_index2 = -1</td>
<td></td>
</tr>
<tr>
<td>static char[]</td>
<td>_itoa_buffer</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_operation = OPERATION_DRAW</td>
<td></td>
</tr>
<tr>
<td>static byte[]</td>
<td>_png_index</td>
<td></td>
</tr>
<tr>
<td>static byte[]</td>
<td>_png_result</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_png_size</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_png_start_crc</td>
<td></td>
</tr>
<tr>
<td>static ASprite[][]</td>
<td>_poolCacheSprites</td>
<td></td>
</tr>
<tr>
<td>static short[][]</td>
<td>_poolCacheStack</td>
<td>Table stores the indices of module images being cached.</td>
</tr>
<tr>
<td>static int[][]</td>
<td>_poolCacheStackIndex</td>
<td>The loop cursor</td>
</tr>
<tr>
<td>static int[][]</td>
<td>_poolCacheStackMax</td>
<td>The pool size</td>
</tr>
<tr>
<td>static int</td>
<td>_rectX1</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_rectX2</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_rectY1</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_rectY2</td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td>_text_h</td>
<td>Used to gather dimensions of a string. (call UpdateStringSize() to update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this value)</td>
</tr>
<tr>
<td>static int</td>
<td>_text_w</td>
<td>Used to gather dimensions of a string. (call UpdateStringSize() to update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this value)</td>
</tr>
<tr>
<td>static short[]</td>
<td>_warpTextInfo</td>
<td></td>
</tr>
<tr>
<td>static final int</td>
<td>BS_AF_OFF_SHORT = (1 &lt;&lt; 18)</td>
<td></td>
</tr>
<tr>
<td>static final int</td>
<td>BS_ANIMS = (1 &lt;&lt; 16)</td>
<td></td>
</tr>
<tr>
<td>static final int</td>
<td>BS_DEFAULT_DOJA = (BS_MODULES</td>
<td>BS_FRAMES</td>
</tr>
<tr>
<td>static final int</td>
<td>BS_DEFAULT_MIDP1 = (BS_MODULES</td>
<td>BS_MODULES_XY</td>
</tr>
<tr>
<td>static final int</td>
<td>BS_DEFAULT_MIDP1b = (BS_MODULES</td>
<td>BS_FRAMES</td>
</tr>
<tr>
<td></td>
<td>BS_DEFAULT_MIDP1c = (BS_MODULES</td>
<td>BS_MODULES_XY</td>
</tr>
</tbody>
</table>

*BS_DEFAULT_MIDP1c*
static final int BS_FRAMES | BS_ANIMS | BS_SINGLE_IMAGE

static final int BS_DEFAULT_MIDP2 = (BS_MODULES | BS_FRAMES | BS_ANIMS | BS_MODULE_IMAGES)

static final int BS_DEFAULT_NOKIA = (BS_DEFAULT_MIDP2)

static final int BS_FM_OFF_SHORT = (1 << 10)

static final int BS_FM_PALETTE = (1 << 14)

static final int BS_FRAME_COLL_RC = (1 << 13)

static final int BS_FRAME_RECTS = (1 << 15)

static final int BS_FRAMES = (1 << 8)

static final int BS_GIF_HEADER = (1 << 31)

static final int BS_IMAGE_SIZE_INT = (1 << 7)

static final int BS_KEEP_PAL = (1 << 26)

static final int BS_MODULE_IMAGES = (1 << 24)

static final int BS_MODULE_IMAGES_FX = (1 << 23)

static final int BS_MODULE_USAGE = (1 << 30)

static final int BS_MODULES = (1 << 0)

static final int BS_MODULES_IMG = (1 << 2)

static final int BS_MODULES_USAGE = (1 << 6)

static final int BS_MODULES_WH_SHORT = (1 << 4)

static final int BS_MODULES_XY = (1 << 1)

static final int BS_MODULES_XY_SHORT = (1 << 5)

static final int BS_NAF_1_BYTE = (1 << 19)

static final int BS_NFM_1_BYTE = (1 << 11)

static final int BS_NO_AF_START = (1 << 17)

static final int BS_PNG_CRC = (1 << 25)

static final int BS_SKIP_FRAME_RC = (1 << 12)

static final int BS_TRANSP_FIRST = (1 << 27)

static final int BS_TRANSP_LAST = (1 << 28)

static final short BSPRITE_v003 = (short)0x03DF

static final short BSPRITE_v004 = (short)0x04DF

static final short BSPRITE_v005 = (short)0x05DF

static final short currentChunkType

static final short ENCODE_FORMAT_I127RLE = 0x27F1

static final short ENCODE_FORMAT_I16 = 0x1600

static final short ENCODE_FORMAT_I2 = 0x0200

static final short ENCODE_FORMAT_I256 = 0x5602

static final short ENCODE_FORMAT_I256RLE = 0x56F2

static final short ENCODE_FORMAT_I4 = 0x0400

static final short ENCODE_FORMAT_I64RLE = 0x64F0

static final byte FLAG_FLIP_X = 0x01

static final byte FLAG_FLIP_Y = 0x02

static final byte FLAG_HYPER_FM = 0x10
<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td><strong>FLAG_INDEX_EX_MASK</strong> = 0xC0</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>FLAG_OFFSET_AF</strong> = 0x20</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>FLAG_OFFSET_FM</strong> = 0x10</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>FLAG_ROT_90</strong> = 0x04</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>FLAG_USER0</strong> = 0x10</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>FLAG_USER1</strong> = 0x20</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>INDEX_EX_MASK</strong> = 0x0300</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>INDEX_EX_SHIFT</strong> = 2</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>INDEX_MASK</strong> = 0x03FF</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>k_itoa_buffer_size</strong> = 33</td>
</tr>
<tr>
<td>static final byte[]</td>
<td><strong>MAGIC_IDAT_h</strong> = { (byte)0x78, (byte)0x9C, (byte)0x01, }</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MAX_TRANSFORMATION_FLAGS</strong> = 8</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_ARC</strong> = 3</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_FILL_ARC</strong> = 4</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_FILL_RECT</strong> = 2</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_FILL_TRIANGLE</strong> = 7</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_IMAGE</strong> = 0</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_MARKER</strong> = 5</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_RECT</strong> = 1</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>MD_TRIANGLE</strong> = 6</td>
</tr>
<tr>
<td>static int</td>
<td><strong>mem</strong> = 0</td>
</tr>
<tr>
<td>static final int[]</td>
<td><strong>midp2_flags</strong></td>
</tr>
<tr>
<td>static int</td>
<td><strong>mod</strong></td>
</tr>
<tr>
<td>static int</td>
<td><strong>mResizeRef</strong> = <strong>RESIZE_REF_240x320</strong></td>
</tr>
<tr>
<td>static final int</td>
<td><strong>OPERATION_COMPUTERECT</strong> = 1</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>OPERATION_DRAW</strong> = 0</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>OPERATION_MARK</strong> = 3</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>OPERATION_RECORD</strong> = 2</td>
</tr>
<tr>
<td>static final short</td>
<td><strong>PIXEL_FORMAT_0565</strong> = (short)0x6505</td>
</tr>
<tr>
<td>static final short</td>
<td><strong>PIXEL_FORMAT_1555</strong> = (short)0x5515</td>
</tr>
<tr>
<td>static final short</td>
<td><strong>PIXEL_FORMAT_4444</strong> = (short)0x4444</td>
</tr>
<tr>
<td>static final short</td>
<td><strong>PIXEL_FORMAT_8888</strong> = (short)0x8888</td>
</tr>
<tr>
<td>static int</td>
<td><strong>record_frame</strong> = -1</td>
</tr>
<tr>
<td>static int</td>
<td><strong>record_index</strong> = -1</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_CREATERGB</strong> = 1</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_DRAW_ON_MUTABLE</strong> = 2</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_NONE</strong> = 0</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_NOT_CACHED</strong> = 3</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_REF_176x220</strong> = 1</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>RESIZE_REF_240x256</strong> = 2</td>
</tr>
<tr>
<td>Static final int</td>
<td>RESIZE_REF_240x320 = 0</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Static boolean</td>
<td>s_bAspectRatio</td>
</tr>
<tr>
<td>Static boolean</td>
<td>s_bBilinear</td>
</tr>
<tr>
<td>Static byte[]</td>
<td>s_MapChar</td>
</tr>
<tr>
<td>Static int[]</td>
<td>s_rc = new int[4]</td>
</tr>
<tr>
<td>Static int</td>
<td>s_resizeType</td>
</tr>
<tr>
<td>Static final int</td>
<td>SCALE_SHIFT = 16</td>
</tr>
<tr>
<td>Static final short</td>
<td>SUPPORTED_VERSION = BSPRITE_v005</td>
</tr>
<tr>
<td>Static byte</td>
<td>temp_byte []</td>
</tr>
<tr>
<td>Static int</td>
<td>temp_int []</td>
</tr>
<tr>
<td>Static short</td>
<td>temp_short []</td>
</tr>
<tr>
<td>Static int</td>
<td>transform_int []</td>
</tr>
</tbody>
</table>
Detailed Description

**ASprite** Aurora Sprite class.

Implementation for sprites exported by AuroraGT editor. Contains methods to handle displaying of strings.

**Author:**
Ionut Matasaru ([ionut.matasaru@gameloft.com](mailto:ionut.matasaru@gameloft.com))
Dragos Velicu ([dragos.velicu@gameloft.com](mailto:dragos.velicu@gameloft.com))
Constructor & Destructor Documentation

**ASprite** ( ) [package]

Empty constructor. Does nothing.
void BuildAnimCacheImages ( int \textit{palette},
  int \textit{anim} ) [package]

Create cache images for all modules all frame used by a animation.
This funcion will call BuildFrameCacheImages(ASprite sprite, int frame, int palette) for each animation frame.

\textbf{Parameters:}

\begin{itemize}
  \item \textit{palette} Sprite palette index. Will be ignored if the sprite is exported using BS_FM_PALETTE flag and if \texttt{GLLibConfig.sprite\_useFMPalette} is enabled.
  \item \textit{anim} Animation index
\end{itemize}

\textbf{See also:}

\begin{itemize}
  \item \texttt{ASprite.BuildFrameCacheImages(int palette, int frame)}
  \item \texttt{ASprite.BuildCacheImages(int pal, int m1, int m2, int pal\_copy)}
  \item \texttt{GLLibConfig.sprite\_useFMPalette}
  \item \texttt{ASprite.BS\_FM\_PALETTE}
\end{itemize}

void BuildCacheImages ( int \textit{pal},
  int \textit{m1},
  int \textit{m2},
  int \textit{pal\_copy} ) [package]

Build Cache Images for this sprite.

\textbf{Parameters:}
void BuildFrameCacheImages ( int  
   palette, 
   int  frame
)  

Create cache images for all modules used by a frame.

**Parameters:**

- **palette** Sprite palette index. Will be ignored if the sprite is exported using BS_FM_PALETTE flag and if GLLibConfig.sprite_useFMPalette is enabled.
- **frame** Frame index

**See also:**

- ASprite.BuildCacheImages(int pal, int m1, int m2, int pal_copy)
- GLLibConfig.sprite_useFMPalette
- ASprite.BS_FM_PALETTE

int CountFrameModules ( int  
   frame
)  

Count the number of modules in a frame.

**Parameters:**

- **frame** The frame to be examined

**Returns:**

The number of modules in the frame

static int Crc32 ( byte  
   buffer[], 
   int  start, 
   int  count,
)
Crc32.

Parameters:

- buffer: the buffer for which the CRC is computed.
- start: the index in the buffer for which the CRC is computed.
- count: the length to compute CRC.
- crc: ???to be documented???

Returns:

???to be documented???

Object DecodeImage ( int module ) [package]

Decodes loaded module data.

Parameters:

- module: Module to be decoded.

Returns:

Object The decoded image data.

void DecodeImageToByteArray ( byte[] dest, int module, boolean img2dRGBA, boolean half ) [package]

Decodes loaded module data to a byte array.

Parameters:

- dest: The destination array.
- module: The module to be decoded.
- img2dRGBA: True if the decoded image should be RGBA, false if RGB only.
- half: True if the decoded image should be half-scaled.
static void DrawLine ( Graphics g,  
    int  x1,  
    int  y1,  
    int  x2,  
    int  y2,  
    int  color,  
    int  radius,  
    int  initial )

Bresenham's Line-Drawing Algorithm.

Parameters:

  g       The graphics context
  x1      X coordinate at the start of the line
  y1      Y coordinate at the start of the line
  x2      X coordinate at the end of the line
  y2      Y coordinate at the end of the line
  color   The color of the line
  radius  The radius of the line
  initial Initial offset

void DrawNumber ( Graphics g,  
    int  num,  
    int  radix,  
    int  minDigit,  
    int  x,  
    int  y,  
    int  anchor,  
    boolean restorecol )

Draws text given a number.

Parameters:

  g            The graphics context
  num          The number to be drawn
  radix        The radix of number, only support 10 decimal system now
  minDigit     The min digit, not enough digits to make up for 0
  x            The X coordinate to be drawn to
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>The Y coordinate to be drawn to</td>
</tr>
<tr>
<td>anchor</td>
<td>The anchor flags to be used for drawing</td>
</tr>
<tr>
<td>restorecol</td>
<td>Boolean determining if the currently set font palette be restored after this operation</td>
</tr>
</tbody>
</table>

```java
void DrawNumber ( Graphics g,
                int num,
                int minDigit,
                int x,
                int y,
                int anchor )
```

Draws text given a number.

**Parameters:**
- $g$: The graphics context
- $num$: The number to be drawn
- $minDigit$: The min digit, not enough digits to make up for 0
- $x$: The X coordinate to be drawn to
- $y$: The Y coordinate to be drawn to
- $anchor$: The anchor flags to be used for drawing

```java
void DrawPage ( Graphics g,
                String s,
                int x,
                int y,
                int anchor,
                int start,
                int end )
```

Draws a string with appropriate newline characters as a page given a start and end.

**Parameters:**
- $g$: The graphics context
- $s$: The string to be drawn
- $x$: The X coordinate to be drawn to
void DrawPage ( Graphics g,  
    String s, 
    int x,  
    int y,  
    int anchor  
)  
[package]

Draws a string with appropriate newline characters as a page.

Parameters:
- `g` The graphics context
- `s` The string to be drawn
- `x` The X coordinate to be drawn to
- `y` The Y coordinate to be drawn to
- `anchor` The anchor flags to be used for drawing

void DrawPage ( Graphics g,  
    byte[] s,  
    int x,  
    int y,  
    int anchor  
)  
[package]

Draws a byte array containing a string with appropriate newline characters as a page.

Parameters:
- `g` The graphics context
- `s` The byte array to be drawn
- `x` The X coordinate to be drawn to
- `y` The Y coordinate to be drawn to
- `anchor` The anchor flags to be used for drawing
void DrawPageB ( Graphics  $g$,  
        String  $s$,  
        short[]  $info$,  
        int  $x$,  
        int  $y$,  
        int  $startLine$,  
        int  $maxLines$,  
        int  $anchor$  )  

Draws a string with appropriate newline characters and wrapping information as a page.

**Parameters:**

- $g$ The graphics context
- $s$ The string to be drawn
- $info$ The wrapping information for the string
- $x$ The X coordinate to be drawn to
- $y$ The Y coordinate to be drawn to
- $startLine$ The starting line to begin drawing from
- $maxLines$ The maximum number of lines to draw
- $anchor$ The anchor flags to be used for drawing

void DrawString ( Graphics  $g$,  
        String  $s$,  
        int  $x$,  
        int  $y$,  
        int  $anchor$,  
        boolean  $restorecol$  )  

Draws text given a string.

**Parameters:**

- $g$ The graphics context
- $s$ The string to be drawn
- $x$ The X coordinate to be drawn to
- $y$ The Y coordinate to be drawn to
- $anchor$ The anchor flags to be used for drawing
- $restorecol$ Boolean determining if the currently set font palette be restored after this
void DrawString ( Graphics g, byte[] bs, int x, int y, int anchor, boolean restorecol )

Draws text given a byte array containing a string.

Parameters:

- g The graphics context
- bs A byte array containing a string
- x The X coordinate to be drawn to
- y The Y coordinate to be drawn to
- anchor The anchor flags to be used for drawing
- restorecol Boolean determining if the currently set font palette be restored after this operation

void DrawString ( Graphics g, String s, int x, int y, int anchor )

Draws text given a string.

Parameters:

- g The graphics context
- s The string to be drawn
- x The X coordinate to be drawn to
- y The Y coordinate to be drawn to
- anchor The anchor flags to be used for drawing
void DrawString ( Graphics  \textit{g},
\text{byte[]}  \textit{bs},
int  \textit{x},
int  \textit{y},
int  \textit{anchor}
) \quad [\text{package}]

Draws text given a byte array containing a string.

\textbf{Parameters:}

\begin{itemize}
  \item \textit{g} The graphics context
  \item \textit{bs} A byte array containing a string
  \item \textit{x} The X coordinate to be drawn to
  \item \textit{y} The Y coordinate to be drawn to
  \item \textit{anchor} The anchor flags to be used for drawing
\end{itemize}

void DrawStringOrChars ( Graphics  \textit{g},
\text{String}  \textit{s},
\text{char[]}  \textit{charBuff},
int  \textit{x},
int  \textit{y},
int  \textit{anchor},
boolean  \textit{restorecol}
) \quad [\text{package}]

void FreeAnimCacheImages ( int  \textit{palette},
int  \textit{anim}
) \quad [\text{package}]

Free cache images for all modules of all frames used by a animation.

This function will call \texttt{FreeFrameCacheImages(int palette, int frame)} for each animation frame.

\textbf{Parameters:}

\begin{itemize}
  \item \textit{palette} Sprite palette index. Will be ignored if the sprite is exported using BS_FM_PALETTE flag and if \texttt{GLLibConfig.sprite_useFMPalette} is enabled.
  \item \textit{anim} Animation index
\end{itemize}
Free all cached data.

Free cache images for all modules used by a frame.

**Parameters:**
- **palette**  Sprite palette index. Will be ignored if the sprite is exported using BS_FM_PALETTE flag and if `GLLibConfig.sprite_useFMPalette` is enabled.
- **frame**  Frame index

See also:

- `ASprite.FreeModuleImage(int nPal, int nMod)`
- `GLLibConfig.sprite_useFMPalette`
- `ASprite.BS_FM_PALETTE`

Free data after the modules have been cached.
Frees a cached module.

**Parameters:**

- `nPal` The palette index
- `nMod` The module to be freed (-1 for all modules in this palette)

---

Generates a palette based on another palette.

**Parameters:**

- `type` Palette type (PAL_ORIGINAL, PAL_INVISIBLE, PAL_RED_YELLOW, PAL_BLUE_CYAN, PAL_GREEN, PAL_GREY or PAL_BLEND_BLACK)
- `pal` The existing palette

**Returns:**
The generated palette

---

Generates a palette based on another palette.

**Parameters:**

- `type` Palette type (PAL_ORIGINAL, PAL_INVISIBLE, PAL_RED_YELLOW, PAL_BLUE_CYAN, PAL_GREEN, PAL_GREY or PAL_BLEND_BLACK)
- `pal` The existing palette

**Returns:**
The generated palette
int GetAFrameFlags ( int anim,
               int aframe
          ) [package]

Gets the flags associated with a frame in an animation.

**Parameters:**

- `anim` The animation to be examined
- `aframe` The frame in the animation

**Returns:**

the flags associated with this frame in the animation

void GetAFrameRect ( int anim,
               int aframe,
               int rectIndex,
               int[] rect,
               int flags
          ) [package]

Get a FrameRect from a specific anim and aframe.

**Parameters:**

- `anim` anim to query for Frame Rect.
- `aframe` aframe to query for Frame Rect.
- `rectIndex` which rect to get for this frame.
- `rect` int array to hlod the rect, [x, y, w, h].
- `flags` flip x/y flags

void GetAFrameRect ( int[] rc,
               int anim,
               int aframe,
               int posX,
               int posY,
               int flags
          ) [package]

Gets the rectangle occupied by an animation frame if it were to be drawn at posX/posY with flags.
Parameters:

- \textit{rc}  The rectangle array to be populated
- \textit{anim}  The animation containing the frame
- \textit{aframe}  The frame to be examined
- \textit{posX}  The X coordinate to be considered
- \textit{posY}  The Y coordinate to be considered
- \textit{flags}  The flags to be considered

\begin{verbatim}
int GetAFrames ( int \textit{anim} ) [package]
\end{verbatim}

Get the number of frames in an animation.

Parameters:

- \textit{anim}  The animation to be examined

Returns:

The number of frames

\begin{verbatim}
int GetAFramesOX ( int \textit{v} ) [package]
\end{verbatim}

To be documented.

\begin{verbatim}
.. \end{verbatim}

Parameters:

- \textit{v}

Returns:

Int

\begin{verbatim}
int GetAFramesOY ( int \textit{v} ) [package]
\end{verbatim}

To be documented.

\begin{verbatim}
.. \end{verbatim}

Parameters:
Returns: Int

int GetAFrameTime ( int anim, int aframe ) [package]

Gets the time of a frame in an animation.

Parameters:
  
  * anim The animation to be examined
  * aframe The frame in the animation

Returns: The frame time

int GetAnimFrame ( int anim, int aframe ) [package]

Gets the ID of a frame inside an animation.

Parameters:
  
  * anim The animation to examine
  * aframe The frame inside the animation

Returns: The frame ID

boolean GetBold ( ) [package]

Gets a value that indicates whether this Font is Bold.

Returns: The font is Bold
int GetCharFrame ( int c ) [package]

byte [] GetCharMap ( ) [package]

Returns the current character map used by DrawString.

Returns: byte array containg the the current character map

short [][] GetCharMapShort ( ) [package]

new text rendering

static int GetChars ( char[] charBuf, int i, int radix, int minDigit ) [static, package]

int getCharSize ( char c ) [package]

Gets the width of the given character using this sprite.

Parameters:
  c  The character to be examined

Returns: The width of the character in pixels

int GetCharSpacing ( ) [package]

Gets the current character spacing for the sprite.
### Returns:
The character spacing

```c
int GetCurrentMMapping(
)[package]
```

Gets the current Module mapping index.

### Returns:
The index

```c
int GetCurrentPalette(
)[package]
```

Gets the current palette for the sprite @ returns the current palette.

### Returns:
The current palette

```c
static int GetCurrentStringHeight(
)[static, package]
```

Gets the current string height used by DrawString.

### Returns:
The current string height

```c
static int GetCurrentStringWidth(
)[static, package]
```

Gets the current string width used by DrawString.

### Returns:
The current string width

```c
int GetFModuleOX(int v)
)[package]
```

To be documented.

..
int GetFModuleOY (int v)

To be documented.

Parameters:

Returns:

Int

void GetFModuleRect (int[] rc, int frame, int fmodule, int posX, int posY, int flags, int hx, int hy)

Gets the rectangle occupied by a module in a frame if it were to be drawn at posX/posY with flags.

Parameters:

rc The rectangle array to be populated
frame The frame containing the module
fmodule The module to be examined
posX The X coordinate to be considered
posY The Y coordinate to be considered
flags The flags to be considered
**int GetFModules ( int frame ) [package]**

Get the number of modules in a frame.

**Parameters:**

- *frame* The frame to be examined

**Returns:**

The number of modules

**int GetFontHeight ( ) [package]**

Gets the font height of the sprite.

**Returns:**

Int font height

**int GetFrameCount ( ) [package]**

Get the number of frames in the sprite.

**Returns:**

the number of frames

**int GetFrameHeight ( int frame ) [package]**

Gets the height of a frame.

**Parameters:**

- *frame* The frame to be examined

**Returns:**
### int [] GetFrameMarkers ( int frame ) [package]

**Get all the marker module positions of a frame.**

**Parameters:**
- `frame` frame to query for markers.

**Returns:**
- an int array with all the markers or null if there are no maker on this frame.

### int GetFrameModule ( int frame, int fmodule ) [package]

**Get the id of a module in a frame.**

**Parameters:**
- `frame` the frame containing the module
- `fmodule` the module to be examined

**Returns:**
- the module id

### int GetFrameModuleFlags ( int frame, int fmodule ) [package]

**Get the flags set on a module in a frame.**

**Parameters:**
- `frame` the frame containing the module
- `fmodule` the module to be examined

**Returns:**
the flags set on the module

```c
int GetFrameModuleHeight ( int frame,
                            int fmodule )  [package]
```

Get the height of a module inside a frame.

**Parameters:**
- `frame` : The frame to be examined
- `fmodule` : The module inside the frame

**Returns:**
- The height of the module

```c
int GetFrameModulePalette ( int frame,
                            int fmodule )  [package]
```

Retrieve a frame module's palette, if the sprite is exported using BS_FM_PALETTE flag.

If this flag is not used or if `GLLibConfig.sprite_useFMPalette` isn't used, returns current sprite palette.

**Parameters:**
- `frame` : Frame index
- `fmodule` : Frame module index

**Returns:**
- Frame module palette

**See also:**
- `GLLibConfig.sprite_useFMPalette`
- `ASprite.BS_FM_PALETTE`
- `ASprite.GetCurrentPalette()`
int GetFrameModuleWidth ( int frame, int fmodule ) [package]

Get the width of a module inside a frame.

Parameters:
  frame The frame to be examined
  fmodule The module inside the frame

Returns:
The width of the module

int GetFrameModuleX ( int frame, int fmodule ) [package]

Gets the X coordinate of a module inside a frame.

Parameters:
  frame The frame to be examined
  fmodule The module inside the frame

Returns:
The X coordinate inside the frame

int GetFrameModuleY ( int frame, int fmodule ) [package]

Gets the Y coordinate of a module inside a frame.

Parameters:
  frame The frame to be examined
  fmodule The module inside the frame

Returns:
void GetFrameRect ( int   frame,
    int   rectIndex,
    int[] rect,
    int   flags
  )  [package]

Get a FrameRect from a specific frame.

Parameters:
  frame    frame to query for Frame Rect.
  rectIndex which rect to get for this frame.
  rect     int array to hold the rect, [x, y, w, h].
  flags    flip x/y flags

void GetFrameRect ( int[] rc,
    int   frame,
    int   posX,
    int   posY,
    int   flags
  )  [package]

Gets the rectangle occupied by a frame if it were to be drawn at posX=posY with flags.

Parameters:
  rc     The rectangle array to be populated
  frame  The frame to be examined
  posX   The X coordinate to be considered
  posY   The Y coordinate to be considered
  flags  The flags to be considered
Gets the rectangle occupied by a frame if it were to be drawn at posX/posY with flags.

**Parameters:**
- `rc` The rectangle array to be populated
- `frame` The frame to be examined
- `posX` The X coordinate to be considered
- `posY` The Y coordinate to be considered
- `flags` The flags to be considered
- `hx` ??? Unused ???
- `hy` ??? Unused ???

Get The FrameRect count for a specific frame.

**Parameters:**
- `frame` frame to query for Frame Rect Count.

**Returns:**
- the Frame rect count for this frame.

Gets the number of frames in the sprite.

**Returns:**
- The number of frames

Gets the width of a frame.
Parameters:

frame  The frame to be examined

Returns:
The frame width

int GetLineHeight ( ) [package]

Gets the current line height of the sprite.

Returns:
The line height

int GetLineSpacing ( ) [package]

Returns the current line spacing for the sprite.

Returns:
The current line spacing

int GetModuleCount ( ) [package]

Get the number of modules in the sprite.

Returns:
the number of modules

Object GetModuleData ( int nModule, int nPalette ) [package]

Get the Module's Data of a specific Palette.

Parameters:
nModule  Module number
Returns:
Object (int[], short[] ...) of the module/palette or null if nModule/nPalette are not valid or if the Image does not exist.

int GetModuleHeight ( int module ) [package]

Gets the height of a module.

Parameters:
module The module to be examined

Returns:
The height

int GetModuleHeightOrg ( int module ) [package]

To be documented.

Parameters:
module

Returns:
Int

Image GetModuleImage ( int nModule, int nPalette ) [package]

Get the Module's Image of a specific Palette.

Parameters:
nModule Module number
nPalette Palette number
Returns:
Image of the module/palette or null if nModule/nPalette are not valid or if the Image does not exist.

```c
void GetModuleRect ( int[] rc,
                      int module,
                      int posX,
                      int posY,
                      int flags )
```

Gets the rectangle occupied by a module if it were to be drawn at posX/posY with flags.

Parameters:
- `rc` The rectangle array to be populated
- `module` The module to be examined
- `posX` The X coordinate to be considered
- `posY` The Y coordinate to be considered
- `flags` The flags to be considered

```c
int GetModuleWidth ( int module )
```

Gets the width of a module.

Parameters:
- `module` The module to be examined

Returns:
The width

```c
int GetModuleWidthOrg ( int module )
```

To be documented.
### Parameters:

- *module*

### Returns:

- Int

### int GetModuleX ( int *module* ) [package]

Gets the x coordinate of a module.

### Parameters:

- *module*  The module to be examined

### Returns:

- The x coordinate

### int GetModuleY ( int *module* ) [package]

Gets the y coordinate of a module.

### Parameters:

- *module*  The module to be examined

### Returns:

- The y coordinate

### Object GetPalette ( int *nPalette* ) [package]

Gets a palette array.

### Parameters:

- *nPalette*  Palette index

### Returns:

- The palette array object
int GetSpaceWidth ( ) [package]

Gets the current space width of the sprite.

**Returns:**

The space width

boolean GetUnderline ( ) [package]

Gets a value that indicates whether this Font is underlined.

**Returns:**

The font is underlined

static void InitCachePool ( int poolCount ) [static]

Initialize Cache Pool.

**Parameters:**

*poolCount* init number of pool

static void InitPoolSize ( int poolIndex, int size ) [static]

Initialize pool size.

**Parameters:**

*poolIndex* index of pool

*size* size of pool

void Load ( byte[] file, int offset, int pal_flags, int tr_flags, ...
Load the sprite from the given byte array and associated image.

Parameters:

- **file**: A byte array containing a sprite file
- **offset**: The offset inside the array at which to start reading
- **pal_flags**: Palette flags
- **tr_flags**: Transformation flags
- **sprImage**: The image associated with the sprite

```c
void Load (byte[] file, int offset, Image sprImage) {
}
```

Load the sprite from the given byte array with flags.

Parameters:

- **file**: A byte array containing a sprite file
- **offset**: The offset inside the array at which to start reading
- **pal_flags**: Palette flags
- **tr_flags**: Transformation flags

```c
void Load (byte[] file, int offset, int pal_flags, int tr_flags) {
}
```
void Load ( byte[] file, int offset )

Load the sprite from the given byte array.

Parameters:

  file     A byte array containing a sprite file
  offset   The offset inside the array at which to start reading

void ModifyPalette ( int palNb, int color )

Modify palette, original palette will turn into the alpha value, and set one and only color for all palette original palette = 0x00RRGGBB, color parameter = 0x00CCCCCC result palette = 0xBBCCCCCC.

Parameters:

  palNb   The palette number
  color   The color to be set

void ModifyPaletteAlpha ( int palNb, int alpha )

Modify the alpha value of a palette.

Parameters:

  palNb   The palette number
  alpha   The alpha to be set

void ModifyPaletteAlphaUsingAltPalette ( int p_iPaletteID, int p_iAlphaPaletteID )

[package]
Modify palette, original palette will have its alpha channel set using the second palette's RED channel.

Original palette = 0x00RRGGBB, Alpha palette = 0x00AA0000 Result palette = 0xAARRGGBB

**Parameters:**

- `p_iPaletteID` is the index of the palette you want to alter
- `p_iAlphaPaletteID` is the index of the palette that holds the alpha channel in its RED channel

```c
void ModifyPaletteAlphaUsingLastPalette ( int p_iPaletteID )
```

This is just a wrapper around ModifyPaletteAlphaUsingAltPalette which automatically uses the last palette for the alpha information.

**Parameters:**

- `p_iPaletteID` is the index of the palette you want to alter

```c
void PaintAFrame ( Graphics g,
       int anim,
       int aframe,
       int posX,
       int posY,
       int flags,
       int hx,
       int hy )
```

Draws an animation frame atPosX/PosY with Flags.

**Parameters:**

- `g` The Graphics context
- `anim` The animation containing the frame
- `aframe` The frame to be drawn
- `posX` The X coordinate to be drawn to
- `posY` The Y coordinate to be drawn to
- `flags` The flags to be used for this operation
- `hx` The height of the area to be drawn
hy  The width of the area to be drawn

```c
void PaintAFrame ( Graphics  g,
       int        anim,
       int        aframe,
       int        posX,
       int        posY,
       int        flags
           )  [package]
```

Draws an animation frame at PosX/PosY with Flags.

**Parameters:**
- `g`  The Graphics context
- `anim`  The animation containing the frame
- `aframe`  The frame to be drawn
- `posX`  The X coordinate to be drawn to
- `posY`  The Y coordinate to be drawn to
- `flags`  The flags to be used for this operation

```c
void PaintFModule ( Graphics  g,
       int        frame,
       int        fmodule,
       int        posX,
       int        posY,
       int        flags,
       int        hx,
       int        hy
           )  [package]
```

Draws a module from a frame at PosX/PosY with Flags.

**Parameters:**
- `g`  The Graphics context
- `frame`  The frame containing the module
- `fmodule`  The module to be drawn
- `posX`  The X coordinate to be drawn to
- `posY`  The Y coordinate to be drawn to
- `flags`  The flags to be used for this operation
void PaintFModule ( Graphics g, 
    int frame, 
    int fmodule, 
    int posX, 
    int posY, 
    int flags )

[package]

Draws a module from a frame at posX/posY with Flags.

Parameters:

- g: The Graphics context
- frame: The frame containing the module
- fmodule: The module to be drawn
- posX: The X coordinate to be drawn to
- posY: The Y coordinate to be drawn to
- flags: The flags to be used for this operation

void PaintFrame ( Graphics g, 
    int frame, 
    int posX, 
    int posY, 
    int flags, 
    int hx, 
    int hy )

[package]

Draws a frame at posX/posY with Flags.

Parameters:

- g: The Graphics context
- frame: The frame to be drawn
- posX: The X coordinate to be drawn to
- posY: The Y coordinate to be drawn to
- flags: The flags to be used for this operation
- hx: The height of the area to be drawn
- hy: The width of the area to be drawn
The width of the area to be drawn

```c
void PaintFrame ( Graphics g,
    int  frame,
    int  posX,
    int  posY,
    int  flags
) {
    [package]
}
```

Draws a frame atPosX/PosY with Flags.

**Parameters:**
- `g` The Graphics context
- `frame` The frame to be drawn
- `posX` The X coordinate to be drawn to
- `posY` The Y coordinate to be drawn to
- `flags` The flags to be used for this operation

```c
void PaintModule ( Graphics g,
    int  module,
    int  posX,
    int  posY,
    int  flags
) {
    [package]
}
```

Draws a module atPosX/PosY with Flags.

**Parameters:**
- `g` The Graphics context
- `module` The frame to be drawn
- `posX` The X coordinate to be drawn to
- `posY` The Y coordinate to be drawn to
- `flags` The flags to be used for this operation

```c
void PaintPrecomputedFrame ( Graphics g,
    int  x,
    int  y,
    int  frame
) {
    [package]
}
```
Paints a pre-computed frame at x, y.

**Parameters:**
- `g` The graphics context
- `x` The X coordinate to draw to
- `y` The Y coordinate to draw to
- `frame` The frame to draw

```c
void PrecomputeAllFrames ( Graphics g )
```

Precomputes all frames.

**Parameters:**
- `g` The graphics context

```c
void PrecomputeFrame ( Graphics g, int frame, int flags )
```

Precompute a frame.

**Parameters:**
- `g` The graphics context
- `frame` The frame to be precomputed
- `flags` The drawing flags to be used

```c
static void ResetCachePool ( int poolIndex )
```

Release pool.

**Parameters:**
- `poolIndex` index of pool
void ResizeCoords() {
    // To be documented...
}

int scaleX(int x) {
    // To be documented.
    // Parameters:
    //   x
}

int scaleY(int y) {
    // To be documented.
    // Parameters:
    //   y
}

void SetBold(boolean bBold) [package]

    // Sets the bold for the font.
    // Parameters:
    //   bBold  the bold flag

void SetCharMap(short[] pNewMap) [package]

    // Sets the character map used by DrawString for the new text rendering code.
    // Parameters:
**void SetCharMap ( byte[] pNewMap ) [package]**

Sets the character map used by DrawString.

**Parameters:**
- *pNewMap*  byte array containing the character map to be set

**static void SetCharMapStatic ( byte[] pNewMap ) [static, package]**

Sets the character map used by DrawString from a static context.

**Parameters:**
- *pNewMap*  byte array containing the character map to be set

**void SetCharSpacing ( int spacing ) [package]**

Sets the character spacing for the sprite.

**Parameters:**
- *spacing*  the character spacing

**void SetCharSpacingToDefault ( ) [package]**

Reverts the character spacing to default for the sprite.

**void SetCurrentMMapping ( int map ) [package]**

Sets the current Module mapping index.

**Parameters:**
- *map*  The index to be set
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void SetCurrentPalette</code></td>
<td>(int <code>pal</code>)</td>
<td>Sets the current palette for the sprite.</td>
</tr>
<tr>
<td><strong>Parameters:</strong></td>
<td></td>
<td><strong>Parameters:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>pal</code> the palette to be set</td>
</tr>
<tr>
<td><code>void SetDefaultFontMetrics</code></td>
<td>( )</td>
<td>Sets the default values for font ascent, descent, height, space, and line spacing, based on the data found in the sprite file.</td>
</tr>
<tr>
<td><code>static void SetGraphics</code></td>
<td>(Graphics <code>g1</code>)</td>
<td>Sets the current graphics context.</td>
</tr>
<tr>
<td><strong>Parameters:</strong></td>
<td></td>
<td><strong>Parameters:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>g1</code> The graphics context to be set</td>
</tr>
<tr>
<td><code>void SetLineHeight</code></td>
<td>(int <code>nHeight</code>)</td>
<td>Sets the line height for the sprite.</td>
</tr>
<tr>
<td><strong>Parameters:</strong></td>
<td></td>
<td><strong>Parameters:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>nHeight</code> the line height</td>
</tr>
<tr>
<td><code>void SetLineHeightToDefault</code></td>
<td>( )</td>
<td>Reverts the line height to the default for the sprite.</td>
</tr>
<tr>
<td><code>void SetLineSpacing</code></td>
<td>(int <code>spacing</code>)</td>
<td>Sets the line spacing for the sprite.</td>
</tr>
</tbody>
</table>
Parameters:

- spacing the line spacing to be set

```c
void SetLineSpacingToDefault() {
    Reverts line spacing to the default for the sprite.
}
```

```c
void SetModuleImage(Image pImg, int nModule, int nPalette) {
    Set the Module's Image of a specific Palette.
}
```

**Parameters:**

- pImg Image to set in the ASprite
- nModule Module number
- nPalette Palette number

```c
void SetModuleImagesArray(Object pData) {
    Sets the module cached image array.
}
```

**Parameters:**

- pData image array

```c
void SetModuleMapping(int map, byte[] mmp) {
    Set the module mapping at an index.
}
```

**Parameters:**

- map The mapping index
- mmp The module mapping
void SetPool ( int poolIndex )

set cache pool of ASprite

Parameters:
  poolIndex  index of pool

void SetResizeParameters ( int spriteId,
                          int resizeMode,
                          boolean correctY )

To be documented.

Parameters:
  spriteId
  resizeMode
  correctY

void SetSpaceWidth ( int spacing ) [package]

Sets the space width of the sprite.

Parameters:
  spacing  the space width to set

void SetSpaceWidthToDefault ( ) [package]

Reverts the space width to the default for the sprite.

static void SetSubString ( int i1,
                          int i2

Set the current substring.

**Parameters:**
- $i_1$ index 1
- $i_2$ index 2

```java
static void SetTempBuffer ( Object pArray )
```

Sets the temporary buffer array to be used for decoding operations.

**Parameters:**
- $pArray$ The buffer array to be used

```java
void SetUnderline ( boolean bUnderline )
```

Sets the underline for the font.

**Parameters:**
- $bUnderline$ the underline flag

```java
static int StringTokenize ( String s,
        int index1,
        int index2,
        char token,
        int[] indices
    )
```

Tokenize a string given a separating token character.

**Parameters:**
- $s$ The string to be tokenized
- $index1$ The character position to begin tokenizing
- $index2$ The character position to stop tokenizing
- $token$ The token character to be used
- $indices$ an int array where the tokenization configuration is to be stored
Returns:
The number of lines resulting from tokenization

String TextFitToFixedWidth ( String str, int text_w ) [package]

Fits the supplied string to a given width.

Parameters:
str The string to be wrapped
text_w The width to be wrapped to

Returns:
The wrapped string

byte [] TextFitToFixedWidth ( byte[] str, int text_w ) [package]

Fits the supplied byte array containing a string to a given width.

Parameters:
str The string byte array to be wrapped
text_w The width to be wrapped to

Returns:
The wrapped byte array

int [] TransformRGB ( int[] image_data, int sizeX, int sizeY, int flags ) [package]

Applies transformations to an RGB buffer.
Parameters:
- `image_data` RGB buffer to transform
- `sizeX` width of the rgb data
- `sizeY` height of the rgb data
- `flags` Transformation flags

```java
void unload () [package]
```

unload the sprite

```java
void UpdateNumberSize ( int num,
                         int radix,
                         int minDigit
                       ) [package]
```

Updates the text size for the sprite given a number.

Parameters:
- `num` The number to be drawn
- `radix` The radix of number
- `minDigit` The min digit, not enough digits to make up for 0

```java
void UpdateStringOrCharsSize ( String s,
                                char[] charBuff
                              ) [package]
```

```java
void UpdateStringSize ( String s ) [package]
```

Updates the current string size for the sprite given a string.

Parameters:
- `s` the string

```java
void UpdateStringSize ( byte[] bs ) [package]
```
Updates the current string size for the sprite given a byte array containing a string.

**Parameters:**

- `bs` a byte array containing a string

```java
short [] WrapTextB ( String s,
                   int width,
                   int height )
```

Wraps the provided string to a given area and returns an array containing the wrapped configuration.

**Parameters:**

- `s` The string to be wrapped
- `width` The width of the area to wrap to
- `height` The height of the area to wrap to

**Returns:**

Short array containing the wrapping configuration
**Member Data Documentation**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte []</td>
<td><em>aframes</em></td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_aframes_flags</td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_aframes_frame</td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_aframesอก_byte</td>
<td>[package]</td>
</tr>
<tr>
<td>short []</td>
<td>_aframesอก_short</td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_aframes_y_byte</td>
<td>[package]</td>
</tr>
<tr>
<td>short []</td>
<td>_aframes_y_short</td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_aframes_time</td>
<td>[package]</td>
</tr>
<tr>
<td>boolean</td>
<td>_alpha</td>
<td>[package]</td>
</tr>
<tr>
<td>short []</td>
<td>_anims_af_start</td>
<td>[package]</td>
</tr>
<tr>
<td>byte []</td>
<td>_anims_naf</td>
<td>[package]</td>
</tr>
</tbody>
</table>
int _cur_pal [package]

int _cur_pool = -1 [package]

which pool the cached images will be created in */

short data_format [package]

byte [] _fmodules [package]

byte [] _fmodules_flags [package]

byte [] _fmodules_id [package]

byte [] _fmodules_ox_byte [package]

short [] _fmodules_ox_short [package]

byte [] _fmodules_oy_byte [package]

short [] _fmodules_oy_short [package]

byte [] _fmodules_pal [package]
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[] _frames_col</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>short[] _frames_col_short</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>short[] _frames_fm_start</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>byte[] _frames_nfm</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>byte[] _frames_rc</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>short[] _frames_rc_short</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>byte[] _frames_rects</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>short[] _frames_rects_short</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>short[] _frames_rects_start</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>byte[][] _gifHeader</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>Graphics _graphics</td>
<td>static, package</td>
<td></td>
</tr>
<tr>
<td>int[] _header_size</td>
<td>package</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>_i64rle_color_bits</code></td>
<td><code>int</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_i64rle_color_mask</code></td>
<td><code>int</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_images_count</code></td>
<td><code>int</code></td>
<td>[static, package]</td>
</tr>
<tr>
<td><code>_images_size</code></td>
<td><code>int</code></td>
<td>[static, package]</td>
</tr>
<tr>
<td><code>_index1</code></td>
<td><code>int</code></td>
<td>= -1 [static, package]</td>
</tr>
<tr>
<td><code>_index2</code></td>
<td><code>int</code></td>
<td>= -1 [static, package]</td>
</tr>
<tr>
<td><code>itoa_buffer</code></td>
<td><code>char[]</code></td>
<td>[static, package]</td>
</tr>
<tr>
<td><code>_main_image</code></td>
<td><code>Image[]</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_map</code></td>
<td><code>short[][]</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_module_colors_byte</code></td>
<td><code>byte[]</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_module_colors_int</code></td>
<td><code>int[]</code></td>
<td>[package]</td>
</tr>
<tr>
<td><code>_module_image_imageAA</code></td>
<td><code>Image[][]</code></td>
<td>[package]</td>
</tr>
<tr>
<td>Type</td>
<td>Package</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>_module_image_intAAA</td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td>_module_types</td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td>_modules_data</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>_modules_data_off_int</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_data_off_short</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_extra_info</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_extra_pointer</td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td>_modules_h_byte</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_h_scaled</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_h_short</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>_modules_image_shortAAA</td>
<td></td>
</tr>
</tbody>
</table>
byte [] _modules_usage [package]

byte [] _modules_w_byte [package]

short [] _modules_w_scaled [package]

short [] _modules_w_short [package]

byte [] _modules_x_byte [package]

short [] _modules_x_short [package]

byte [] _modules_y_byte [package]

short [] _modules_y_short [package]

int _nModules [package]

int _old_pal [package]

int _operation = OPERATION_DRAW [static, package]

byte [] _pal_data [package]
<table>
<thead>
<tr>
<th>Data Type</th>
<th>Array</th>
<th>Declaration</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>[] _pal_int</td>
<td>[package]</td>
</tr>
<tr>
<td>short</td>
<td>[] _pal_short</td>
<td>[package]</td>
</tr>
<tr>
<td>int</td>
<td>_palettes</td>
<td>[package]</td>
</tr>
<tr>
<td>byte</td>
<td>[] _png_index</td>
<td>[static, package]</td>
</tr>
<tr>
<td>int</td>
<td>[] _PNG_packed_IDAT_ADLER</td>
<td>[package]</td>
</tr>
<tr>
<td>int</td>
<td>[] _PNG_packed_IDAT_CRC</td>
<td>[package]</td>
</tr>
<tr>
<td>int</td>
<td>[] _PNG_packed_IHDR_CRC</td>
<td>[package]</td>
</tr>
<tr>
<td>int</td>
<td>[] _PNG_packed_PLTE_CRC</td>
<td>[package]</td>
</tr>
<tr>
<td>int</td>
<td>[] _PNG_packed_tRNS_CRC</td>
<td>[package]</td>
</tr>
<tr>
<td>byte</td>
<td>[] _png_result</td>
<td>[static, package]</td>
</tr>
<tr>
<td>int</td>
<td>_png_size</td>
<td>[static, package]</td>
</tr>
<tr>
<td>int</td>
<td>_png_start_crc</td>
<td>[static, package]</td>
</tr>
</tbody>
</table>
### poolCacheSprites

[static, package]

ASprite

### poolCacheStack

[static, package]

short

table stores the indices of module images being cached.

### poolCacheStackIndex

[static, package]

int

the loop cursor

### poolCacheStackMax

[static, package]

int

the pool size

### rectX1

[static, package]

int

### rectX2

[static, package]

int

### rectY1

[static, package]

int

### rectY2

[static, package]

int

### text_h

[static, package]

int

Used to gather dimensions of a string. (call **UpdateStringSize()** to update this value).
<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>int_text_w</td>
<td>Used to gather dimensions of a string. Call <code>UpdateStringSize()</code> to update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>this value.</td>
</tr>
<tr>
<td>byte</td>
<td>[] _transp</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>[] _w_pos</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>[] _warpTextInfo</td>
<td></td>
</tr>
<tr>
<td>final</td>
<td>int BASE = 65521</td>
<td>Static value representing a constant.</td>
</tr>
<tr>
<td>final</td>
<td>int BLOCK_INFO_SIZE = 11</td>
<td>Static value representing the size of a block.</td>
</tr>
<tr>
<td>final</td>
<td>int BS_AF_OFF_SHORT = (1 &lt;&lt; 18)</td>
<td>Static value representing a flag.</td>
</tr>
<tr>
<td>final</td>
<td>int BS_ANIMS = (1 &lt;&lt; 16)</td>
<td>Static value representing a flag.</td>
</tr>
<tr>
<td>final</td>
<td>int BS_DEFAULT_DOJA = (BS_MODULES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BS_FRAMES</td>
</tr>
<tr>
<td>final</td>
<td>int BS_DEFAULT_MIDP1 = (BS_MODULES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BS_MODULES_XY</td>
</tr>
<tr>
<td>final</td>
<td>int BS_DEFAULT_MIDP1b = (BS_MODULES</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BS_MODULES</td>
</tr>
</tbody>
</table>
final int BS_DEFAULT_MIDP1c = (BS_MODULES | BS_MODULES_XY | BS_FRAMES | BS_ANIMS) [static, package]

final int BS_DEFAULT_MIDP2 = (BS_MODULES | BS_FRAMES | BS_ANIMS | BS_MODULE_IMAGES) [static, package]

final int BS_DEFAULT_NOKIA = (BS_DEFAULT_MIDP2) [static, package]

final int BS_FM_OFF_SHORT = (1 << 10) [static, package]

final int BS_FM_PALETTE = (1 << 14) [static, package]

final int BS_FRAME_COLL_RC = (1 << 13) [static, package]

final int BS_FRAME_RECTS = (1 << 15) [static, package]

final int BS_FRAMES = (1 << 8) [static, package]

final int BS_GIF_HEADER = (1 << 31) [static, package]

final int BS_IMAGE_SIZE_INT = (1 << 7) [static, package]

final int BS_KEEP_PAL = (1 << 26) [static, package]

final int BS_MODULE_IMAGES = (1 << 24) [static, package]
final int BS_MODULE_IMAGES_FX = (1 << 23) [static, package]
final int BS_MODULE_USAGE = (1 << 30) [static, package]
final int BS_MODULES = (1 << 0) [static, package]
final int BS_MODULES_IMG = (1 << 2) [static, package]
final int BS_MODULES_USAGE = (1 << 6) [static, package]
final int BS_MODULES_WH_SHORT = (1 << 4) [static, package]
final int BS_MODULES_XY = (1 << 1) [static, package]
final int BS_MODULES_XY_SHORT = (1 << 5) [static, package]
final int BS_NAF_1_BYTE = (1 << 19) [static, package]
final int BS_NFM_1_BYTE = (1 << 11) [static, package]
final int BS_NO_AF_START = (1 << 17) [static, package]
final int BS_PNG_CRC = (1 << 25) [static, package]
final int BS_SINGLE_IMAGE = (1 << 29) [static, package]

final int BS_SKIP_FRAME_RC = (1 << 12) [static, package]

final int BS_TRANSP_FIRST = (1 << 27) [static, package]

final int BS_TRANSP_LAST = (1 << 28) [static, package]

final short BSPRITE_v003 = (short)0x03DF [static, package]

final short BSPRITE_v004 = (short)0x04DF [static, package]

final short BSPRITE_v005 = (short)0x05DF [static, package]

final int CRC32_POLYNOMIAL = 0xEDB88320 [static]

int crcTable[] = new int[256] [static]

int currentChunkType [static, package]

final short ENCODE_FORMAT_I127RLE = 0x27F1 [static, package]

final short ENCODE_FORMAT_I16 = 0x1600 [static, package]
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final short ENCODE_FORMAT_I2</code></td>
<td>0x0200</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final short ENCODE_FORMAT_I256</code></td>
<td>0x5602</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final short ENCODE_FORMAT_I256RLE</code></td>
<td>0x56F2</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final short ENCODE_FORMAT_I4</code></td>
<td>0x0400</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final short ENCODE_FORMAT_I64RLE</code></td>
<td>0x64F0</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_FLIP_X</code></td>
<td>0x01</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_FLIP_Y</code></td>
<td>0x02</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_HYPER_FM</code></td>
<td>0x10</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final int FLAG_INDEX_EX_MASK</code></td>
<td>0xC0</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_OFFSET_AF</code></td>
<td>0x20</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_OFFSET_FM</code></td>
<td>0x10</td>
<td>static, package</td>
</tr>
<tr>
<td><code>final byte FLAG_ROT_90</code></td>
<td>0x04</td>
<td>static, package</td>
</tr>
</tbody>
</table>
final byte `FLAG_USER0` = 0x10 [static, package]

final byte `FLAG_USER1` = 0x20 [static, package]

final int `HEADER_LEVEL0_MAX_WBITS` = 30938 [static]

int `hRef` [package]

int `hTarget` [package]

final byte `IDAT[]` = {'I', 'D', 'A', 'T'} [static, protected]

final byte `IEND[]` = {'I', 'E', 'N', 'D'} [static, protected]

final byte `IHDR[]` = {'I', 'H', 'D', 'R'} [static, protected]

final int `INDEX_EX_MASK` = 0x0300 [static, package]

final int `INDEX_EX_SHIFT` = 2 [static, package]

final int `INDEX_MASK` = 0x03FF [static, package]

final byte `INFO32[]` = {8, 6, 0, 0} [static, protected]
### Import Statements

- `final byte INFO8[] = { 8, 3, 0, 0 } [static, protected]`
- `final int k_itoa_buffer_size = 33 [static, package]`

### Comments

All possible chars for representing a number as a String only support 10 decimal system now.

- `final byte MAGIC[] = {-119, 80, 78, 71, 13, 10, 26, 10 } [static, protected]`
- `final byte[] MAGIC_IDAT_h = { (byte)0x78, (byte)0x9C, (byte)0x01, } [static, package]`
- `final byte[] MAGIC_IEND = {0x00, 0x00, 0x00, 0x00, 0x49, 0x45, 0x4e, 0x44, (byte)0xae, 0x42, 0x60, (byte)0x82}`

### Constants

- `final int MAX_TRANSFORMATION_FLAGS = 8 [static, package]`
- `final int MD_ARC = 3 [static, package]`
- `final int MD_FILL_ARC = 4 [static, package]`
- `final int MD_FILL_RECT = 2 [static, package]`
- `final int MD_FILL_TRIANGLE = 7 [static, package]`
- `final int MD_IMAGE = 0 [static, package]`
final int MD_MARKER = 5 [static, package]

final int MD_RECT = 1 [static, package]

final int MD_TRIANGLE = 6 [static, package]

int mem = 0 [static, package]

final int[] midp2_flags [static, package]

Initial value:

```
{
    javax.microedition.lcdui.game.Sprite.TRANS_NONE,
    javax.microedition.lcdui.game.Sprite.TRANS_MIRROR,
    javax.microedition.lcdui.game.Sprite.TRANS_MIRROR_ROT180,
    javax.microedition.lcdui.game.Sprite.TRANS_ROT180,
    javax.microedition.lcdui.game.Sprite.TRANS_ROT90,
    javax.microedition.lcdui.game.Sprite.TRANS_MIRROR_ROT90,
    javax.microedition.lcdui.game.Sprite.TRANS_MIRROR_ROT270,
    javax.microedition.lcdui.game.Sprite.TRANS_ROT270,
}
```

int mod [static, package]

boolean mResizeCorrectY = false [package]

int mResizeRef = RESIZE_REF_240x320 [static, package]

final int NMAX = 5552 [static]
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPERATION_COMPUTERECT</td>
<td>1</td>
</tr>
<tr>
<td>OPERATION_DRAW</td>
<td>0</td>
</tr>
<tr>
<td>OPERATION_MARK</td>
<td>3</td>
</tr>
<tr>
<td>OPERATION_RECORD</td>
<td>2</td>
</tr>
<tr>
<td>PAL_BLEND_BLACK</td>
<td>5</td>
</tr>
<tr>
<td>PAL_BLUE_CYAN</td>
<td>2</td>
</tr>
<tr>
<td>PAL_GREEN</td>
<td>3</td>
</tr>
<tr>
<td>PAL_GREY</td>
<td>4</td>
</tr>
<tr>
<td>PAL_INVISIBLE</td>
<td>0</td>
</tr>
<tr>
<td>PAL_ORIGINAL</td>
<td>-1</td>
</tr>
<tr>
<td>PAL_RED_YELLOW</td>
<td>1</td>
</tr>
<tr>
<td>PIXEL_FORMAT_0565</td>
<td>(short)0x6505</td>
</tr>
<tr>
<td>Constant Name</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>final short <strong>PIXEL_FORMAT_1555</strong></td>
<td>(short)0x5515</td>
</tr>
<tr>
<td>final short <strong>PIXEL_FORMAT_4444</strong></td>
<td>(short)0x4444</td>
</tr>
<tr>
<td>final short <strong>PIXEL_FORMAT_8888</strong></td>
<td>(short)0x8888</td>
</tr>
<tr>
<td>final byte <strong>PLTE[]</strong></td>
<td>{'P', 'L', 'T', 'E'}</td>
</tr>
<tr>
<td>final int <strong>PNG_INFO_SIZE</strong></td>
<td>57</td>
</tr>
<tr>
<td>int <strong>record_frame</strong></td>
<td>-1</td>
</tr>
<tr>
<td>int <strong>record_index</strong></td>
<td>-1</td>
</tr>
<tr>
<td>final int <strong>RESIZE_CREATERGB</strong></td>
<td>1</td>
</tr>
<tr>
<td>final int <strong>RESIZE_DRAW_ON_MUTABLE</strong></td>
<td>2</td>
</tr>
<tr>
<td>final int <strong>RESIZE_NONE</strong></td>
<td>0</td>
</tr>
<tr>
<td>final int <strong>RESIZE_NOT_CACHED</strong></td>
<td>3</td>
</tr>
<tr>
<td>final int <strong>RESIZE_REF_176x220</strong></td>
<td>1</td>
</tr>
</tbody>
</table>
final int RESIZE_REF_240x256 = 2 [static, package]

final int RESIZE_REF_240x320 = 0 [static, package]

boolean s_bAspectRatio [static, package]

boolean s_bBilinear [static, package]

byte [] s_MapChar [static, package]

int [] s_rc = new int[4] [static, package]

int s_resizeType [static, package]

final int SCALE_SHIFT = 16 [static, package]

final short SUPPORTED_VERSION = BSPRITE_v005 [static, package]

byte temp_byte[] [static, package]

int temp_int[] [static, package]

short temp_short[] [static, package]
<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>transform_int</code></td>
<td><code>int[]</code></td>
<td>Static, package</td>
</tr>
<tr>
<td><code>tRNS</code></td>
<td><code>byte[]</code></td>
<td>Static, protected</td>
</tr>
<tr>
<td><code>wRef</code></td>
<td><code>int</code></td>
<td>Package</td>
</tr>
<tr>
<td><code>wTarget</code></td>
<td><code>int</code></td>
<td>Package</td>
</tr>
<tr>
<td><code>xRatio</code></td>
<td><code>int</code></td>
<td>Package</td>
</tr>
<tr>
<td><code>yRatio</code></td>
<td><code>int</code></td>
<td>Package</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:32 2008 for GLLib by doxygen 1.5.2
Canvas Class Reference

Inherited by GLLib.
GLKey Interface Reference

[GLKey]

Game key code interface. More...

List of all members.
## Static Public Attributes

<table>
<thead>
<tr>
<th>Static final byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>k_consoleLogEnable = k_pound</code></td>
<td>Constant to toggle the Log Console.</td>
</tr>
<tr>
<td><code>k_down = 2</code></td>
<td>Constant for the DOWN key press.</td>
</tr>
<tr>
<td><code>k_dummy = 0</code></td>
<td>Constant for a Dummy key. Every invalid key press will be triggered as dummy.</td>
</tr>
<tr>
<td><code>k_fire = 5</code></td>
<td>Constant for the FIRE / ENTER key press.</td>
</tr>
<tr>
<td><code>k_left = 3</code></td>
<td>Constant for the LEFT key press.</td>
</tr>
<tr>
<td><code>k_menuBack = 19</code></td>
<td>Constant for the Back Softkey press.</td>
</tr>
<tr>
<td><code>k_menuOK = 18</code></td>
<td>Constant for the OK Softkey press.</td>
</tr>
<tr>
<td><code>k_nbKey = 20</code></td>
<td>Constant for the key count.</td>
</tr>
<tr>
<td><code>k_num0 = 6</code></td>
<td>Constant for the Number 0 key press.</td>
</tr>
<tr>
<td><code>k_num1 = 7</code></td>
<td>Constant for the Number 1 key press.</td>
</tr>
<tr>
<td><code>k_num2 = 8</code></td>
<td>Constant for the Number 2 key press.</td>
</tr>
<tr>
<td><code>k_num3 = 9</code></td>
<td>Constant for the Number 3 key press.</td>
</tr>
<tr>
<td><code>k_num4 = 10</code></td>
<td>Constant for the Number 4 key press.</td>
</tr>
<tr>
<td><code>k_num5 = 11</code></td>
<td>Constant for the Number 5 key press.</td>
</tr>
<tr>
<td><code>k_num6 = 12</code></td>
<td>Constant for the Number 6 key press.</td>
</tr>
<tr>
<td><code>k_num7 = 13</code></td>
<td>Constant for the Number 7 key press.</td>
</tr>
<tr>
<td><code>k_num8 = 14</code></td>
<td>Constant for the Number 8 key press.</td>
</tr>
<tr>
<td><code>k_num9 = 15</code></td>
<td>Constant for the Number 9 key press.</td>
</tr>
<tr>
<td>static final byte</td>
<td>k_pound = 17</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>Constant for the # key press.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final byte</th>
<th>k_right = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant for the RIGHT key press.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final byte</th>
<th>k_rmsLogEnable = k_star</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant to toggle the Log in RMS.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final byte</th>
<th>k_star = 16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant for the * key press.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final byte</th>
<th>k_up = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constant for the UP key press.</td>
</tr>
</tbody>
</table>
# Static Package Attributes

<table>
<thead>
<tr>
<th>static final byte</th>
<th><code>k_invalid</code> = -1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Constant for an Invalid key.</em></td>
</tr>
</tbody>
</table>
Detailed Description

Game key code interface.
### Member Data Documentation

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>final byte k_consoleLogEnable = k_pound</td>
<td>static</td>
<td>Constant to toggle the Log Console. Note: May work only in debug.</td>
</tr>
<tr>
<td>final byte k_down = 2</td>
<td>static</td>
<td>Constant for the DOWN key press.</td>
</tr>
<tr>
<td>final byte k_dummy = 0</td>
<td>static</td>
<td>Constant for a Dummy key. Every invalid key press will be triggered as dummy.</td>
</tr>
<tr>
<td>final byte k_fire = 5</td>
<td>static</td>
<td>Constant for the FIRE / ENTER key press.</td>
</tr>
<tr>
<td>final byte k_invalid = -1</td>
<td>static, package</td>
<td>Constant for an Invalid key.</td>
</tr>
<tr>
<td>final byte k_left = 3</td>
<td>static</td>
<td>Constant for the LEFT key press.</td>
</tr>
<tr>
<td>final byte k_menuBack = 19</td>
<td>static</td>
<td></td>
</tr>
</tbody>
</table>
Constant for the Back Softkey press.

**Note:**
This key is handled as immediately released eg: no accumulation is possible.

```
final byte k_menuOK = 18 [static]
```

Constant for the OK Softkey press.

**Note:**
This key is handled as immediately released eg: no accumulation is possible.

```
final byte k_nbKey = 20 [static]
```

Constant for the key count.

```
final byte k_num0 = 6 [static]
```

Constant for the Number 0 key press.

```
final byte k_num1 = 7 [static]
```

Constant for the Number 1 key press.

```
final byte k_num2 = 8 [static]
```

Constant for the Number 2 key press.

```
final byte k_num3 = 9 [static]
```
Constant for the Number 3 key press.

```java
final byte k_num4 = 10 [static]
```

Constant for the Number 4 key press.

```java
final byte k_num5 = 11 [static]
```

Constant for the Number 5 key press.

```java
final byte k_num6 = 12 [static]
```

Constant for the Number 6 key press.

```java
final byte k_num7 = 13 [static]
```

Constant for the Number 7 key press.

```java
final byte k_num8 = 14 [static]
```

Constant for the Number 8 key press.

```java
final byte k_num9 = 15 [static]
```

Constant for the Number 9 key press.

```java
final byte k_pound = 17 [static]
```

Constant for the # key press.
<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>k_right</code></td>
<td>4</td>
<td>Constant for the RIGHT key press.</td>
</tr>
<tr>
<td><code>k_rmsLogEnable</code></td>
<td><code>k_star</code></td>
<td>Constant to toggle the Log in RMS. Note: May work only in debug.</td>
</tr>
<tr>
<td><code>k_star</code></td>
<td>16</td>
<td>Constant for the * key press.</td>
</tr>
<tr>
<td><code>k_up</code></td>
<td>1</td>
<td>Constant for the UP key press.</td>
</tr>
</tbody>
</table>
GLLang Interface Reference

[GLLang]

language code More...

List of all members.
### Static Public Attributes

<table>
<thead>
<tr>
<th>static final int</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>5</td>
<td>Constant for the BR (brazilian) language.</td>
</tr>
<tr>
<td>CN</td>
<td>8</td>
<td>Constant for the CN (Chinese) language.</td>
</tr>
<tr>
<td>CZ</td>
<td>13</td>
<td>Constant for the CZ (Czech) language.</td>
</tr>
<tr>
<td>DE</td>
<td>1</td>
<td>Constant for the DE (german) language.</td>
</tr>
<tr>
<td>ES</td>
<td>4</td>
<td>Constant for the ES (spanish) language.</td>
</tr>
<tr>
<td>FR</td>
<td>2</td>
<td>Constant for the FR (french) language.</td>
</tr>
<tr>
<td>IT</td>
<td>3</td>
<td>Constant for the IT (italian) language.</td>
</tr>
<tr>
<td>JP</td>
<td>7</td>
<td>Constant for the JP (japanese) language.</td>
</tr>
<tr>
<td>KR</td>
<td>9</td>
<td>Constant for the KR (Korean) language.</td>
</tr>
<tr>
<td>PL</td>
<td>12</td>
<td>Constant for the PL (Polish) language.</td>
</tr>
<tr>
<td>PT</td>
<td>6</td>
<td>Constant for the PT (portuguese) language.</td>
</tr>
<tr>
<td>RU</td>
<td>10</td>
<td>Constant for the RU (Russian Federation) language.</td>
</tr>
<tr>
<td>TR</td>
<td>11</td>
<td>Constant for the TR (Turkish) language.</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>EN</strong> = 0</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>Constant for the EN (english) language.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

language code
<table>
<thead>
<tr>
<th>Constant</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR = 5</td>
<td>Brazilian</td>
</tr>
<tr>
<td>CN = 8</td>
<td>Chinese</td>
</tr>
<tr>
<td>CZ = 13</td>
<td>Czech</td>
</tr>
<tr>
<td>DE = 1</td>
<td>German</td>
</tr>
<tr>
<td>EN = 0</td>
<td>English</td>
</tr>
<tr>
<td>ES = 4</td>
<td>Spanish</td>
</tr>
<tr>
<td>FR = 2</td>
<td>French</td>
</tr>
</tbody>
</table>
final int IT = 3 [static]

Constant for the IT (Italian) language.

final int JP = 7 [static]

Constant for the JP (Japanese) language.

final int KR = 9 [static]

Constant for the KR (Korean) language.

final int PL = 12 [static]

Constant for the PL (Polish) language.

final int PT = 6 [static]

Constant for the PT (Portuguese) language.

final int RU = 10 [static]

Constant for the RU (Russian Federation) language.

final int TR = 11 [static]

Constant for the TR (Turkish) language.
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabetical List</td>
<td>Class List</td>
<td>Class Hierarchy</td>
<td>Class Members</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The **GLLib** class is the main class to do a game creation at Gameloft. [More...]

Inherits **Canvas**, and **Runnable**.

**List of all members.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void keyPressed(int keyCode)</code></td>
<td>Key pressed method.</td>
</tr>
<tr>
<td><code>void keyReleased(int keyCode)</code></td>
<td>Key released.</td>
</tr>
<tr>
<td><code>static void Game_KeyClearKeyCode()</code></td>
<td>Clear all the key code. No key input will be valid until you define new key association using <code>Game_keySetKeyCode</code>.</td>
</tr>
<tr>
<td><code>static void Game_KeySetKeyCode(boolean gameAction, int keyCode, int key)</code></td>
<td>Set the keycode for a specific key</td>
</tr>
<tr>
<td><code>static int IsAnyKeyDown()</code></td>
<td>Is there any key that is being held.</td>
</tr>
<tr>
<td><code>static boolean IsKeyDown(int keyFlag)</code></td>
<td>Query a key to know if its down/pressed.</td>
</tr>
<tr>
<td><code>static boolean IsKeyUp(int keyFlag)</code></td>
<td>Query a key to know if its up/unpressed.</td>
</tr>
<tr>
<td><code>static void ResetAKey(int keyFlag)</code></td>
<td>Reset one key to unpressed state.</td>
</tr>
<tr>
<td><code>static void ResetKey()</code></td>
<td>Reset all keys to unpressed.</td>
</tr>
<tr>
<td><code>static int WasAnyKeyPressed()</code></td>
<td>Was there any key just pressed.</td>
</tr>
<tr>
<td><code>static int WasAnyKeyReleased()</code></td>
<td>Was there any key just released.</td>
</tr>
<tr>
<td><code>static boolean WasKeyPressed(int keyFlag)</code></td>
<td>Was there a key just pressed.</td>
</tr>
<tr>
<td><code>static boolean WasKeyReleased(int keyFlag)</code></td>
<td>Was there a key just released.</td>
</tr>
</tbody>
</table>
## Public Member Functions

<table>
<thead>
<tr>
<th>void</th>
<th>Game_Run () throws Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Function to be implemented in every game.</em></td>
</tr>
<tr>
<td>InputStream</td>
<td>GetResourceAsStream (String s)</td>
</tr>
<tr>
<td>GLLib</td>
<td>Constructor.</td>
</tr>
<tr>
<td>void</td>
<td>hideNotify ()</td>
</tr>
<tr>
<td></td>
<td><em>Override of Canvas.hideNotify().</em></td>
</tr>
<tr>
<td>void</td>
<td>paint (Graphics _g)</td>
</tr>
<tr>
<td></td>
<td><em>Standard rendering function.</em></td>
</tr>
<tr>
<td>void</td>
<td>run ()</td>
</tr>
<tr>
<td></td>
<td><em>Game engine main loop/thread.</em></td>
</tr>
<tr>
<td>void</td>
<td>showNotify ()</td>
</tr>
<tr>
<td></td>
<td><em>Override of Canvas.showNotify().</em></td>
</tr>
<tr>
<td>void</td>
<td>sizeChanged (int w, int h)</td>
</tr>
<tr>
<td></td>
<td><em>Called when the drawable area has been changed.</em></td>
</tr>
</tbody>
</table>
### Static Public Member Functions

<table>
<thead>
<tr>
<th>Category</th>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| static final void | `drawPartialRGB`        | (Graphics `g`, int screenWidth, int screenHeight, int rgbData[], int scanlength, int srcX, int srcY, int x, int y, int width, int height, boolean processAlpha)  
Same as `drawPartialRGB` but it's useful when you draw in a back buffer that is bigger than the screen. |
| static final void | `drawPartialRGB`        | (Graphics `g`, int rgbData[], int scanlength, int srcX, int srcY, int x, int y, int width, int height, boolean processAlpha)  
Same as `drawRGB` but it draws safely (without drawing outside the screen). |
| static void       | `LZMA_Inflate`          | (byte[] `compressDat`) throws Exception                                                                                                                                                              |
| static void       | `Vibrate`               | (int `duration`)                                                                                                                                                                                      |

*Make the phone vibrate or flash the back light is the phone has this functionnality.*
## Static Public Attributes

<table>
<thead>
<tr>
<th>static javax.microedition.lcdui.Graphics g = null</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics context where all rendering operation will happen.</td>
</tr>
</tbody>
</table>
## Protected Member Functions

<table>
<thead>
<tr>
<th>Image</th>
<th>GetSoftwareDoubleBuffer ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphics</td>
<td>GetSoftwareDoubleBufferGraphics ()</td>
</tr>
<tr>
<td>void</td>
<td>Init ()</td>
</tr>
<tr>
<td></td>
<td><em>Initialize and start the game engine.</em></td>
</tr>
<tr>
<td>void</td>
<td>Pause ()</td>
</tr>
<tr>
<td></td>
<td><em>Pause the game engine.</em></td>
</tr>
<tr>
<td>void</td>
<td>Quit ()</td>
</tr>
<tr>
<td></td>
<td><em>Request to quit the game engine.</em></td>
</tr>
<tr>
<td>void</td>
<td>Resume ()</td>
</tr>
<tr>
<td></td>
<td><em>Resume the game engine.</em></td>
</tr>
<tr>
<td>void</td>
<td>SetupDisplay ()</td>
</tr>
<tr>
<td></td>
<td><em>Setup the display.</em></td>
</tr>
<tr>
<td>void</td>
<td>UnInit ()</td>
</tr>
<tr>
<td></td>
<td><em>Quit game engine, free all memory and object.</em></td>
</tr>
</tbody>
</table>
Static Protected Member Functions

static void SetupDefaultKey ()

Setup the default key association for this device.
## Package Functions

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>CheckAndDumpConfig ()</code></td>
<td></td>
</tr>
<tr>
<td>abstract void</td>
<td><code>Game_update ()</code> throws Exception</td>
<td><em>Function to be implemented in every game.</em></td>
</tr>
<tr>
<td>String</td>
<td><code>Text_GetLanguageAsString (int languageCode)</code></td>
<td><em>get language code as a string</em></td>
</tr>
<tr>
<td>int</td>
<td><code>Text_GetPhoneDefaultLanguage ()</code></td>
<td><em>get phone default language, if unable to get it, return <code>GLLang.EN</code> (considered as default language)</em></td>
</tr>
</tbody>
</table>
## Static Package Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[static initializer]</strong></td>
<td></td>
</tr>
<tr>
<td>static void <strong>Assert</strong> (boolean test, String errMessage)</td>
<td>Assertion test.</td>
</tr>
<tr>
<td>static final void <strong>ClipRect</strong> (int x, int y, int width, int height)</td>
<td>Intersects the current clip with the specified rectangle.</td>
</tr>
<tr>
<td>static String <strong>ConvertFixedPointToString</strong> (int value, int precision)</td>
<td>Converts a <strong>GLLib</strong> Fixed Point number to a string in base 10.</td>
</tr>
<tr>
<td>static final void <strong>CopyArea</strong> (int x_src, int y_src, int width, int height, int x_dest, int y_dest, int anchor)</td>
<td>Copies the contents of a rectangular area (x_src, y_src, width, height) to a destination area, whose anchor point identified by anchor is located at (x_dest, y_dest).</td>
</tr>
<tr>
<td>static void <strong>Dbg</strong> (String log)</td>
<td>Log a debug string on the console.</td>
</tr>
<tr>
<td>static final void <strong>DrawArc</strong> (int x, int y, int width, int height, int startAngle, int arcAngle)</td>
<td>Draws the outline of a circular or elliptical arc covering the specified rectangle, using the current color and stroke style.</td>
</tr>
<tr>
<td>static final void <strong>DrawChar</strong> (char character, int x, int y, int anchor)</td>
<td>Draws the specified character using the current font and color.</td>
</tr>
<tr>
<td>static final void <strong>DrawChars</strong> (char[] data, int offset, int length, int x, int y, int anchor)</td>
<td>Draws the specified characters using the current font and color.</td>
</tr>
<tr>
<td>static final void <strong>DrawImage</strong> (javax.microedition.lcdui.Image img, int x, int y, int anchor)</td>
<td>Draws the specified image by using the anchor point.</td>
</tr>
<tr>
<td>static final void <strong>DrawLine</strong> (int x1, int y1, int x2, int y2)</td>
<td>Draws a line between the coordinates (x1,y1) and (x2,y2) using the current color and stroke style.</td>
</tr>
<tr>
<td>static final void <strong>DrawRect</strong> (int x, int y, int width, int height)</td>
<td>Draws the outline of the specified rectangle using the current color and stroke style.</td>
</tr>
<tr>
<td>static final void <strong>DrawRegion</strong> (javax.microedition.lcdui.Image src, int x_src, int y_src, int width, int height, int transform, int</td>
<td></td>
</tr>
</tbody>
</table>
Copies a region of the specified source image to a location within the destination, possibly transforming (rotating and reflecting) the image data using the chosen transform function.

static final void DrawRGB (int[] rgbData, int offset, int scanlength, int x, int y, int width, int height, boolean processAlpha)
Renders a series of device-independent RGB+transparency values in a specified region.

static final void DrawRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)
Draws the outline of the specified rounded corner rectangle using the current color and stroke style.

static final void DrawString (String str, int x, int y, int anchor)
Draws the specified String using the current font and color.

static final void DrawSubstring (String str, int offset, int len, int x, int y, int anchor)
Draws the specified String using the current font and color.

static final void FillArc (int x, int y, int width, int height, int startAngle, int arcAngle)
Fills a circular or elliptical arc covering the specified rectangle.

static final void FillRect (int x, int y, int width, int height)
Fills the specified rectangle with the current color.

static final void FillRoundRect (int x, int y, int width, int height, int arcWidth, int arcHeight)
Fills the specified rounded corner rectangle with the current color.

static final void FillTriangle (int x1, int y1, int x2, int y2, int x3, int y3)
Fills the specified triangle with the current color.

static final void Gc ()
call the garbage collector to free memory

static final int GetBlueComponent ()
Gets the blue component of the current color.

static final int GetClipHeight ()
Gets the height of the current clipping area.

static final int GetClipWidth ()
Gets the width of the current clipping area.

static final int GetClipX ()
Gets the X offset of the current clipping area, relative to the coordinate system origin of this graphics context.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int GetClipY ()</td>
<td>Gets the Y offset of the current clipping area, relative to the coordinate system origin of this graphics context.</td>
</tr>
<tr>
<td>static final int GetColor ()</td>
<td>Gets the current color.</td>
</tr>
<tr>
<td>static final int GetDisplayColor (int color)</td>
<td>Gets the color that will be displayed if the specified color is requested.</td>
</tr>
<tr>
<td>static final javax.microedition.lcdui.Font GetFont ()</td>
<td>Gets the current font.</td>
</tr>
<tr>
<td>static final long GetFrameTime ()</td>
<td>get time for this frame (real time when this frame started)</td>
</tr>
<tr>
<td>static final int GetGrayScale ()</td>
<td>Gets the current grayscale value of the color being used for rendering operations.</td>
</tr>
<tr>
<td>static final int GetGreenComponent ()</td>
<td>Gets the green component of the current color.</td>
</tr>
<tr>
<td>static String GetMIME (int idx)</td>
<td>Get the MIME type as a String for a MIME index.</td>
</tr>
<tr>
<td>static int GetNBData ()</td>
<td>Get the number of data in current data file.</td>
</tr>
<tr>
<td>static final long GetRealTime ()</td>
<td>get real time at the moment the function is called</td>
</tr>
<tr>
<td>static final int GetRedComponent ()</td>
<td>Gets the red component of the current color.</td>
</tr>
<tr>
<td>static final int GetScreenHeight ()</td>
<td>get screen height, (if screen orientation changes, it will return the new screen height automatically)</td>
</tr>
<tr>
<td>static final int GetScreenWidth ()</td>
<td>get screen width, (if screen orientation changes, it will return the new screen width automatically)</td>
</tr>
<tr>
<td>static final int GetStrokeStyle ()</td>
<td>Gets the stroke style used for drawing operations.</td>
</tr>
<tr>
<td>static final int GetTranslateX ()</td>
<td>Gets the X coordinate of the translated origin of this graphics context.</td>
</tr>
<tr>
<td>static final int GetTranslateY ()</td>
<td>Gets the Y coordinate of the translated origin of this graphics context.</td>
</tr>
<tr>
<td>static void InitSharedRms (String strVendor, String strMidletName)</td>
<td></td>
</tr>
</tbody>
</table>
Sets vendor name and midlet names - used for shared rms read/write.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static final long Math_Abs (long a)</code></td>
<td>Absolut value of a long.</td>
</tr>
<tr>
<td><code>static final int Math_Abs (int a)</code></td>
<td>Absolut value of an integer.</td>
</tr>
<tr>
<td><code>static int Math_Atan (int dx, int dy)</code></td>
<td>Arctan.</td>
</tr>
<tr>
<td><code>static int Math_AtanSlow (int dx, int dy)</code></td>
<td>Arctan, very slow but accurate method, it find angle by dichotomy.</td>
</tr>
<tr>
<td><code>static void Math_Bezier2D (int x1, int y1, int x2, int y2, int x3, int y3, int interp)</code></td>
<td>Three control point 2D Bezier interpolation.</td>
</tr>
<tr>
<td><code>static void Math_Bezier3D (int x1, int y1, int z1, int x2, int y2, int z2, int x3, int y3, int z3, int interp)</code></td>
<td>Three control point 3D Bezier interpolation.</td>
</tr>
<tr>
<td><code>static int Math_Cos (int angle)</code></td>
<td>Cosinus.</td>
</tr>
<tr>
<td><code>static int Math_DegreeToFixedPointAngle (int a)</code></td>
<td>Convert a degree angle into a fixed point angle.</td>
</tr>
<tr>
<td><code>static int Math_Det (int x1, int y1, int x2, int y2)</code></td>
<td>Compute the determinant of 2 2Dvector.</td>
</tr>
<tr>
<td><code>static int Math_DistPointLine (int x0, int y0, int x1, int y1, int x2, int y2)</code></td>
<td>Compute the distance from a point to a segment/line.</td>
</tr>
<tr>
<td><code>static int Math_Div10 (int a)</code></td>
<td>Divide by 10 a number, with result in fixed point.</td>
</tr>
<tr>
<td><code>static int Math_DotProduct (int x1, int y1, int x2, int y2)</code></td>
<td>Compute the dot product of 2 2Dvector.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Add (int summand1, int summand2)</code></td>
<td>Addition for Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_DegreesToAngleFixedPoint (int angle)</code></td>
<td>Convert an angle from degrees to fixed point angle.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_DegreesToRadians (int angle)</code></td>
<td>Convert an angle from degrees to radians (both fixed point and not, since this is just a ratio).</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Det (int x1, int y1, int x2, int y2)</code></td>
<td>Compute the determinant of 2 2Dvector in Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Divide (int dividend, int divisor)</code></td>
<td>Division for Fixed Point.</td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_DotProduct(int x1, int y1, int x2, int y2)</code></td>
<td>Compute the dot product of 2 2D vectors in Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_LineCircleIntersect(int x1, int y1, int x2, int y2, int circleX, int circleY, int radius)</code></td>
<td>Find intersect points between a line and a circle.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_LineRectangleIntersect(int x1, int y1, int x2, int y2, int rectX, int rectY, int rectW, int rectH)</code></td>
<td>Find intersect points between a line and a rectangle.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Multiply(int multiplicand, int multiplier)</code></td>
<td>Multiplication for Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Norm(int x, int y)</code></td>
<td>Normal of a vector in Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_NormPow(int x, int y)</code></td>
<td>Compute pow of the normal of a vector in Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_PointLineDistance(int lineX1, int lineY1, int lineX2, int lineY2, int pointX, int pointY)</code></td>
<td>Compute the distance from a point to a segment/line.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_RadiansToAngleFixedPoint(int angle)</code></td>
<td>Convert an angle from radians to fixed point angle.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_RadiansToDegrees(int angle)</code></td>
<td>Convert an angle from radians to degrees (both fixed point and not, since this is just a ratio).</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Round(int value)</code></td>
<td>Rounds value to closest whole number in Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Sqrt(long value)</code></td>
<td>Square Root for Long Fixed Point values.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Sqrt(int value)</code></td>
<td>Square Root for Integer Fixed Point values.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Square(int value)</code></td>
<td>Square for Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPoint_Subtract(int minuend, int subtrahend)</code></td>
<td>Subtraction for Fixed Point.</td>
</tr>
<tr>
<td><code>static int Math_FixedPointAdjust(int a)</code></td>
<td>Adjust a fixed point in base 8 to this fixed point base.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>static int Math_FixedPointAngleToDegree(int a)</code></td>
<td>Convert a fixed point angle into a degree angle.</td>
</tr>
<tr>
<td><code>static int Math_FixedPointToInt(int a)</code></td>
<td>Convert a fixed point value to an int value.</td>
</tr>
<tr>
<td><code>static void Math_Init(String packName, int cos, int sqrt)</code></td>
<td>Math initialisation, by reading specified table from a package.</td>
</tr>
<tr>
<td><code>static int Math_IntToFixedPoint(int a)</code></td>
<td>Convert a fixed point value to an int value.</td>
</tr>
<tr>
<td><code>static int Math_Log2(int a)</code></td>
<td>return log base 2</td>
</tr>
<tr>
<td><code>static final long Math_Max(long a, long b)</code></td>
<td></td>
</tr>
<tr>
<td><code>static final int Math_Max(int a, int b)</code></td>
<td></td>
</tr>
<tr>
<td><code>static final long Math_Min(long a, long b)</code></td>
<td></td>
</tr>
<tr>
<td><code>static final int Math_Min(int a, int b)</code></td>
<td></td>
</tr>
<tr>
<td><code>static int Math_Norm(int x, int y, int iter)</code></td>
<td>Norm following newton law approximation.</td>
</tr>
<tr>
<td><code>static int Math_NormPow(int x1, int y1)</code></td>
<td>Compute pow of the normal of a vector.</td>
</tr>
<tr>
<td><code>static void Math_QuickSort(int array[])</code></td>
<td>Quicksort an array of integer.</td>
</tr>
<tr>
<td><code>static int[] Math_QuickSortIndices(int data[], int nbItemPerValue, int itemNb)</code></td>
<td>Get an array of indice corresponding to the sorting of an array of data.</td>
</tr>
<tr>
<td><code>static int[] Math_QuickSortIndices(int data[])</code></td>
<td>Get an array of indice corresponding to the sorting of an array of data.</td>
</tr>
<tr>
<td><code>static void Math_Quiit()</code></td>
<td>Free all math arrays.</td>
</tr>
<tr>
<td><code>static final int Math_Rand()</code></td>
<td>Create a random number.</td>
</tr>
<tr>
<td><code>static int Math_Rand(int a, int b)</code></td>
<td>Create a random int inside the interval [a, b].</td>
</tr>
<tr>
<td><code>static final void Math_RandSetSeed(long seed)</code></td>
<td>set math random seed</td>
</tr>
<tr>
<td><code>static boolean Math_RectIntersect(int Ax0, int Ay0, int Ax1, int Ay1, int Bx0, int By0, int Bx1, int By1)</code></td>
<td>Tell if 2 axis aligned rectangle intersect.</td>
</tr>
<tr>
<td><code>static boolean Math_SameSign(int a, int b)</code></td>
<td>Test 2 numbers to see if they are both of the same</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>Math_SegmentIntersect</code></td>
<td>(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4)</td>
</tr>
<tr>
<td></td>
<td>Tell if 2 segment intersect themselves.</td>
</tr>
<tr>
<td><code>Math_Sin</code></td>
<td>(int a)</td>
</tr>
<tr>
<td></td>
<td>Sinus.</td>
</tr>
<tr>
<td><code>Math_Sqrt</code></td>
<td>(long val)</td>
</tr>
<tr>
<td></td>
<td>Square Root slow.</td>
</tr>
<tr>
<td><code>Math_Sqrt</code></td>
<td>(int x)</td>
</tr>
<tr>
<td></td>
<td>Square Root.</td>
</tr>
<tr>
<td><code>Math_Sqrt_FixedPoint</code></td>
<td>(int val, int precisionLoop)</td>
</tr>
<tr>
<td></td>
<td>Square Root for Fixed Point.</td>
</tr>
<tr>
<td><code>Math_Tan</code></td>
<td>(int angle)</td>
</tr>
<tr>
<td></td>
<td>Tangent.</td>
</tr>
<tr>
<td><code>Mem_ArrayCopy</code></td>
<td>(Object src, int src_position, Object dst, int dst_position, int length)</td>
</tr>
<tr>
<td></td>
<td>throws Exception</td>
</tr>
<tr>
<td></td>
<td>Copies an array from the specified source array, beginning at the specified</td>
</tr>
<tr>
<td></td>
<td>position, to the specified position of the destination array.</td>
</tr>
<tr>
<td><code>Mem_GetArray</code></td>
<td>(byte[] src, int src_off, byte[] dst)</td>
</tr>
<tr>
<td></td>
<td>throws Exception</td>
</tr>
<tr>
<td></td>
<td>fill destination array with content of src array at specified offset</td>
</tr>
<tr>
<td><code>Mem_GetByte</code></td>
<td>(byte[] src, int src_off)</td>
</tr>
<tr>
<td></td>
<td>get a byte value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>Mem_GetInt</code></td>
<td>(byte[] src, int src_off)</td>
</tr>
<tr>
<td></td>
<td>get an int value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>Mem_GetLong</code></td>
<td>(byte[] src, int src_off)</td>
</tr>
<tr>
<td></td>
<td>get a long value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>Mem_GetShort</code></td>
<td>(byte[] src, int src_off)</td>
</tr>
<tr>
<td></td>
<td>get a short value from the specified array at specified offset</td>
</tr>
<tr>
<td><code>Mem_ReadArray</code></td>
<td>(InputStream is)</td>
</tr>
<tr>
<td></td>
<td>Read an array/multiarray, from an inputStream.</td>
</tr>
<tr>
<td><code>Mem_SetArray</code></td>
<td>(byte[] dst, int dst_off, byte[] src)</td>
</tr>
<tr>
<td></td>
<td>throws Exception</td>
</tr>
<tr>
<td></td>
<td>copies whole content of src array in dst array at specified offset</td>
</tr>
<tr>
<td><code>Mem_SetByte</code></td>
<td>(byte[] dst, int dst_off, byte src)</td>
</tr>
<tr>
<td></td>
<td>set a byte value in a byte array at a given offset</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Mem_SetInt</td>
<td>set a int value in a byte array at a given offset</td>
</tr>
<tr>
<td>Mem_SetLong</td>
<td>set a long value in a byte array at a given offset</td>
</tr>
<tr>
<td>Mem_SetShort</td>
<td>set a short value in a byte array at a given offset</td>
</tr>
<tr>
<td>Pack_Close</td>
<td>Close a pack file.</td>
</tr>
<tr>
<td>Pack_CloseShared</td>
<td>throws Exception Closes current pack, previously opened with Pack_OpenShared.</td>
</tr>
<tr>
<td>Pack_LoadMIME</td>
<td>Load the MIME type from a MIME Pack.</td>
</tr>
<tr>
<td>Pack_Open</td>
<td>Open a pack given its path and filename.</td>
</tr>
<tr>
<td>Pack_OpenShared</td>
<td>throws Exception Opens a shared rms pack.</td>
</tr>
<tr>
<td>Pack_PositionAtData</td>
<td>Position the pack pointer to the beginning of data idx.</td>
</tr>
<tr>
<td>Pack_Read</td>
<td>Read a single byte from the current stream.</td>
</tr>
<tr>
<td>Pack_Read16</td>
<td>Read one unsigned short from the current stream.</td>
</tr>
<tr>
<td>Pack_Read32</td>
<td>Read one int from the current stream.</td>
</tr>
<tr>
<td>Pack_ReadArray</td>
<td>Read an array or multiarray from the stream.</td>
</tr>
<tr>
<td>Pack_ReadData</td>
<td>Read and return the data at idx.</td>
</tr>
<tr>
<td>Pack_ReadFully</td>
<td>Read into a byte array.</td>
</tr>
<tr>
<td>Pack_Seed</td>
<td>Set the current offset to addr byte from the beginning.</td>
</tr>
<tr>
<td>Pack_Skip</td>
<td>Skip ahead in the current stream.</td>
</tr>
<tr>
<td>PlatformRequest</td>
<td>Performs platformRequest.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>static void Print(String log)</code></td>
<td>print a string in out stream, does not append character return at the end of line (as opposed to Dbg)</td>
</tr>
<tr>
<td><code>static void Profiler_BeginNamedEvent(String name)</code></td>
<td>Mark the beginning of an event.</td>
</tr>
<tr>
<td><code>static void Profiler_Draw()</code></td>
<td>Draw profiler events to screen.</td>
</tr>
<tr>
<td><code>static void Profiler_End()</code></td>
<td>End capturing events.</td>
</tr>
<tr>
<td><code>static void Profiler_EndNamedEvent()</code></td>
<td>Mark the end of an event.</td>
</tr>
<tr>
<td><code>static void Profiler_Start()</code></td>
<td>Start capturing events.</td>
</tr>
<tr>
<td><code>static byte[] Rms_Read(String strName)</code></td>
<td>Reads data from the recordstore.</td>
</tr>
<tr>
<td><code>static byte[] Rms_ReadShared(String strName, String strVendor, String strMidletName)</code></td>
<td>Read data from shared recordstore.</td>
</tr>
<tr>
<td><code>static void Rms_Write(String strName, byte[] data)</code></td>
<td>Writes data to the recordstore.</td>
</tr>
<tr>
<td><code>static void Rms_WriteShared(String strName, String strVendor, String strMidletName, byte[] data)</code></td>
<td>Write data to shared recordstore.</td>
</tr>
<tr>
<td><code>static void SavePack(String packName, String rmsName)</code></td>
<td>Reads a resource, and writes it into the the recordstore - used for DEBUG purposes.</td>
</tr>
<tr>
<td><code>static final void SetClip(int x, int y, int width, int height)</code></td>
<td>Sets the current clip to the rectangle specified by the given coordinates.</td>
</tr>
<tr>
<td><code>static final void setColor(int red, int green, int blue)</code></td>
<td>Sets the current color to the specified RGB values.</td>
</tr>
<tr>
<td><code>static final void SetColor(int RGB)</code></td>
<td>Sets the current color to the specified RGB values.</td>
</tr>
<tr>
<td><code>static final void SetCurrentGraphics(javax.microedition.lcdui.Image img)</code></td>
<td>set current graphics context -.</td>
</tr>
<tr>
<td><code>static final void SetCurrentGraphics(javax.microedition.lcdui.Graphics graphics)</code></td>
<td>set current graphics context -.</td>
</tr>
<tr>
<td><code>static final void SetFont(javax.microedition.lcdui.Font font)</code></td>
<td>Sets the font for all subsequent text rendering</td>
</tr>
</tbody>
</table>
operations.

static final void SetGrayScale (int value)
Sets the current grayscale to be used for all subsequent rendering operations.

static final void SetStrokeStyle (int style)
Sets the stroke style used for drawing lines, arcs, rectangles, and rounded rectangles.

static int Stream_Read (InputStream is) throws Exception
static int Stream_Read16 (InputStream is) throws Exception
static int Stream_Read32 (InputStream is) throws Exception
static int Stream_ReadFully (InputStream is, byte[] array, int offset, int length)
static void Text_BuildStringCache ()
cache all the text into string Array
static void Text_FreeAll ()
free all the data-array used by the text
static String Text_FromUTF8 (byte[] src, int offset, int len)
get a string from an UTF-8 encoded byte array.
static final int Text_GetNbString ()
get nb of string in current text pack
static String Text_GetString (int index)
get a string given it's index
static void Text_LoadTextFromPack (String filename, int index1, int index2)
open and load text from two text packages inside a pack file
static void Text_LoadTextFromPack (String filename, int index)
static final void Text_SetEncoding (String encoding)
set current text encoding
static final void Translate (int x, int y)
Translates the origin of the graphics context to the point (x, y) in the current coordinate system.
static void Warning (String message)
display a warning
# Static Package Attributes

<table>
<thead>
<tr>
<th>Static final int</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Align</code> = <code>SpecPos + kNumFullDistances - kEndPosModelIndex</code></td>
<td>Constant for positioning the anchor point of text and images below the text or image.</td>
</tr>
<tr>
<td><code>BOTTOM</code> = 32</td>
<td>Constant for positioning the anchor point of text and images below the text or image.</td>
</tr>
<tr>
<td><code>DOTTED</code> = 1</td>
<td>Constant for the DOTTED stroke style.</td>
</tr>
<tr>
<td><code>HCENTER</code> = 1</td>
<td>Constant for centering text and images horizontally around the anchor point.</td>
</tr>
<tr>
<td><code>inputIndex</code></td>
<td></td>
</tr>
<tr>
<td><code>IsMatch</code> = 0</td>
<td></td>
</tr>
<tr>
<td><code>IsRep</code> = <code>IsMatch + (kNumStates &lt;&lt; kNumPosBitsMax)</code></td>
<td></td>
</tr>
<tr>
<td><code>IsRep0Long</code> = <code>IsRepG2 + kNumStates</code></td>
<td></td>
</tr>
<tr>
<td><code>IsRepG0</code> = <code>IsRep + kNumStates</code></td>
<td></td>
</tr>
<tr>
<td><code>IsRepG1</code> = <code>IsRepG0 + kNumStates</code></td>
<td></td>
</tr>
<tr>
<td><code>IsRepG2</code> = <code>IsRepG1 + kNumStates</code></td>
<td></td>
</tr>
<tr>
<td><code>kAlignTableSize</code> = <code>(1 &lt;&lt; kNumAlignBits)</code></td>
<td></td>
</tr>
<tr>
<td><code>kBitModelTotal</code> = <code>(1 &lt;&lt; kNumBitModelTotalBits)</code></td>
<td></td>
</tr>
<tr>
<td><code>kEndPosModelIndex</code> = 14</td>
<td></td>
</tr>
<tr>
<td><code>kLenNumHighBits</code> = 8</td>
<td></td>
</tr>
<tr>
<td><code>kLenNumHighSymbols</code> = <code>(1 &lt;&lt; kLenNumHighBits)</code></td>
<td></td>
</tr>
<tr>
<td><code>kLenNumLowBits</code> = 3</td>
<td></td>
</tr>
<tr>
<td><code>kLenNumLowSymbols</code> = <code>(1 &lt;&lt; kLenNumLowBits)</code></td>
<td></td>
</tr>
<tr>
<td><code>kLenNumMidBits</code> = 3</td>
<td></td>
</tr>
<tr>
<td><code>kLenNumMidSymbols</code> = <code>(1 &lt;&lt; kLenNumMidBits)</code></td>
<td></td>
</tr>
<tr>
<td><code>kMatchMinLen</code> = 2</td>
<td></td>
</tr>
<tr>
<td><code>kNumAlignBits</code> = 4</td>
<td></td>
</tr>
<tr>
<td><code>kNumBitModelTotalBits</code> = 11</td>
<td></td>
</tr>
<tr>
<td><code>kNumFullDistances</code> = <code>(1 &lt;&lt; (kEndPosModelIndex &gt;&gt; 1))</code></td>
<td></td>
</tr>
<tr>
<td><code>kNumLenProbs</code> = <code>LenHigh + kLenNumHighSymbols</code></td>
<td></td>
</tr>
<tr>
<td><code>kNumLenToPosStates</code> = 4</td>
<td></td>
</tr>
<tr>
<td><code>kNumMoveBits</code> = 5</td>
<td></td>
</tr>
<tr>
<td><code>kNumPosBitsMax</code> = 4</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>kNumPosSlotBits</code></td>
<td>6</td>
</tr>
<tr>
<td><code>kNumPosStatesMax</code></td>
<td>$(1 &lt;&lt; kNumPosBitsMax)$</td>
</tr>
<tr>
<td><code>kNumStates</code></td>
<td>12</td>
</tr>
<tr>
<td><code>kNumTopBits</code></td>
<td>24</td>
</tr>
<tr>
<td><code>kStartPosModelIndex</code></td>
<td>4</td>
</tr>
<tr>
<td><code>kTopValue</code></td>
<td>$(1 &lt;&lt; kNumTopBits)$</td>
</tr>
<tr>
<td><code>LEFT</code></td>
<td>4</td>
</tr>
<tr>
<td><code>LenChoice</code></td>
<td>0</td>
</tr>
<tr>
<td><code>LenChoice2</code></td>
<td><code>LenChoice + 1</code></td>
</tr>
<tr>
<td><code>LenCoder</code></td>
<td><code>Align + kAlignTableSize</code></td>
</tr>
<tr>
<td><code>LenHigh</code></td>
<td><code>LenMid + (kNumPosStatesMax &lt;&lt; kLenNumMidBits)</code></td>
</tr>
<tr>
<td><code>LenLow</code></td>
<td><code>LenChoice2 + 1</code></td>
</tr>
<tr>
<td><code>LenMid</code></td>
<td><code>LenLow + (kNumPosStatesMax &lt;&lt; kLenNumLowBits)</code></td>
</tr>
<tr>
<td><code>Literal</code></td>
<td><code>RepLenCoder + kNumLenProbs</code></td>
</tr>
<tr>
<td><code>LZMA_BASE_SIZE</code></td>
<td>1846</td>
</tr>
<tr>
<td><code>LZMA_LIT_SIZE</code></td>
<td>768</td>
</tr>
<tr>
<td><code>LZMA_RESULT_DATA_ERROR</code></td>
<td>1</td>
</tr>
<tr>
<td><code>LZMA_RESULT_NOT_ENOUGH_MEM</code></td>
<td>2</td>
</tr>
<tr>
<td><code>LZMA_RESULT_OK</code></td>
<td>0</td>
</tr>
<tr>
<td><code>m_Buffer</code></td>
<td></td>
</tr>
<tr>
<td><code>m_Code</code></td>
<td></td>
</tr>
<tr>
<td><code>m_current_keys_pressed</code></td>
<td></td>
</tr>
<tr>
<td><code>m_current_keys_released</code></td>
<td></td>
</tr>
<tr>
<td><code>m_current_keys_state</code></td>
<td></td>
</tr>
<tr>
<td><code>m_ExtraBytes</code></td>
<td></td>
</tr>
<tr>
<td><code>m_inSize</code></td>
<td></td>
</tr>
<tr>
<td><code>m_keys_pressed</code></td>
<td><code>Previous frame keys pressed.</code></td>
</tr>
<tr>
<td><code>m_keys_released</code></td>
<td><code>Previous frame keys released.</code></td>
</tr>
<tr>
<td><code>m_keys_state</code></td>
<td><code>Previous frame keys state.</code></td>
</tr>
<tr>
<td><code>m_last_key_pressed</code></td>
<td><code>-9999</code></td>
</tr>
<tr>
<td><code>m_outStream</code></td>
<td></td>
</tr>
<tr>
<td>Static type</td>
<td>Name</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><code>long</code></td>
<td><code>m_Range</code></td>
</tr>
<tr>
<td><code>final int</code></td>
<td><code>Math_Angle180</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_Angle270</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_Angle360</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_Angle90</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_AngleMUL</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_FixedPoint_E</code></td>
</tr>
<tr>
<td></td>
<td><code>Math_FixedPoint_PI</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_INTERSECT_NO_INTERSECT</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_INTERSECT_ONE_POINT</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_INTERSECT_TWO_POINTS</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_SEGMENTINTERSECT_COLLINEAR</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_SEGMENTINTERSECT_DO_INTERSECT</code></td>
</tr>
<tr>
<td></td>
<td><code>MATH_SEGMENTINTERSECT_DONT_INTERSECT</code></td>
</tr>
<tr>
<td><code>byte[][]</code></td>
<td><code>MIME_type</code></td>
</tr>
<tr>
<td></td>
<td><code>PosSlot</code></td>
</tr>
<tr>
<td></td>
<td><code>PROFILER_MAX_EVENTS</code></td>
</tr>
<tr>
<td></td>
<td><code>RepLenCoder</code></td>
</tr>
<tr>
<td></td>
<td><code>RIGHT</code></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>s_application</code></td>
<td>Reference to the application. Usually its a reference to a midlet (or IApplication for doja).</td>
</tr>
<tr>
<td><code>s_display</code></td>
<td>reference to the display</td>
</tr>
<tr>
<td><code>s_game_currentFrameNB</code></td>
<td>Current frame number. Increased every frame.</td>
</tr>
<tr>
<td><code>s_game_FPSAverage</code></td>
<td>Average fps * 100.</td>
</tr>
<tr>
<td><code>s_game_frameDT</code></td>
<td>Delta time between the previous frame and this one.</td>
</tr>
<tr>
<td><code>s_game_interruptNotify</code></td>
<td>Interrupt notifier. Set to true when an interrupt occurred.</td>
</tr>
<tr>
<td><code>s_game_isPaused</code></td>
<td>Pause state of the game. True if the game is paused.</td>
</tr>
<tr>
<td><code>s_game_keyEventIndex</code></td>
<td>Key indice of the key which had an event. -1 if no event.</td>
</tr>
<tr>
<td><code>s_game_keyJustPressed</code></td>
<td>Key indice of the key that was just pressed. -1 if no key was just pressed.</td>
</tr>
<tr>
<td><code>s_game_keyPressedTime</code></td>
<td>Time of the last key press.</td>
</tr>
<tr>
<td><code>s_game_lastFrameTime</code></td>
<td>Similar to <code>s_game_timeWhenFrameStart</code> but internally used for <code>GLLibConfig.useFakeInterruptHandling</code>.</td>
</tr>
<tr>
<td><code>s_game_state</code></td>
<td>Current game state.</td>
</tr>
<tr>
<td><code>s_game_timeWhenFrameStart</code></td>
<td>Current time when the frame started.</td>
</tr>
<tr>
<td><code>s_game_totalExecutionTime</code></td>
<td>Total game execution time.</td>
</tr>
<tr>
<td><code>s_gllib_instance</code></td>
<td>The only instance of this <code>GLLib</code> class.</td>
</tr>
<tr>
<td><code>s_keyLastKeyPressUntranslatedCode</code></td>
<td>Untranslated value of the last keycode. Used before passing through <code>Game_TranslateKeyCode</code>.</td>
</tr>
<tr>
<td><code>s_keyLastKeyStates</code></td>
<td>Last key state obtained through <code>getKeyStates()</code>.</td>
</tr>
<tr>
<td><code>s_keyState[]</code></td>
<td>Current key states bufferised.</td>
</tr>
<tr>
<td><code>s_keyStateRT[]</code></td>
<td></td>
</tr>
</tbody>
</table>
*RealTime key buffer. User should use s_keyState instead.*

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>s_math_bezierX</td>
<td>x coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_math_bezierY</td>
<td>y coordinate of bezier interpolated value through Math_Bezier2D or Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_math_bezierZ</td>
<td>z coordinate of bezier interpolated value through Math_Bezier3D</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_distPointLineX</td>
<td>x coordinate of orthogonal projection of point on a segment using Math_distPointLine</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_distPointLineY</td>
<td>y coordinate of orthogonal projection of point on a segment using Math_distPointLine</td>
</tr>
<tr>
<td>static final int</td>
<td>s_math_F_05</td>
<td>Fixed point value for number 0.5.</td>
</tr>
<tr>
<td>static final int</td>
<td>s_math_F_1</td>
<td>Fixed point value for number 1.</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_intersectPoints[][]</td>
<td>new int[2][2]</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_intersectX</td>
<td>x coordinate of intersection point of 2 segment using Math_segmentIntersect</td>
</tr>
<tr>
<td>static int</td>
<td>s_Math_intersectY</td>
<td>y coordinate of intersection point of 2 segment using Math_segmentIntersect</td>
</tr>
<tr>
<td>static java.util.Random</td>
<td>s_math_random</td>
<td>Random number generator.</td>
</tr>
<tr>
<td>static int</td>
<td>s_pack_lastDataReadMimeType</td>
<td>Mime type of last data that have been read through Pack_Read.</td>
</tr>
<tr>
<td>static final int</td>
<td>SOLID = 0</td>
<td>Constant for the SOLID stroke style.</td>
</tr>
<tr>
<td>static final int</td>
<td>SpecPos = PosSlot + (kNumLenToPosStates &lt;&lt; kNumPosSlotBits)</td>
<td></td>
</tr>
<tr>
<td>static String</td>
<td>text_encoding = &quot;UTF-8&quot;</td>
<td>encoding for text (UTF-8 by default)</td>
</tr>
<tr>
<td>static int</td>
<td>text_nbString</td>
<td>number of string in this text pack</td>
</tr>
<tr>
<td>Static final int</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>tkNumMoveBits = 5</td>
<td>Constant for positioning the anchor point of text and images above the text or image.</td>
<td></td>
</tr>
<tr>
<td>TOP = 16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR = 2</td>
<td>causes the specified image region to be reflected about its vertical center.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT180 = 1</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 180 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT270 = 4</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 270 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT90 = 7</td>
<td>causes the specified image region to be reflected about its vertical center and then rotated clockwise by 90 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_NONE = 0</td>
<td>No transform is applied.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT180 = 3</td>
<td>causes the specified image region to be rotated clockwise by 180 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT270 = 6</td>
<td>causes the specified image region to be rotated clockwise by 270 degrees.</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT90 = 5</td>
<td>causes the specified image region to be rotated clockwise by 90 degrees.</td>
<td></td>
</tr>
<tr>
<td>VCENTER = 2</td>
<td>Constant for centering text and images vertically around the anchor point.</td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

The **GLLib** class is the main class to do a game creation at Gameloft.

The **GLLib** main function is to be the hearth of your game, it implements **Runnable** by starting a Thread. The **GLLib** provides the developer with methods to handle game key events by doing the mapping to your device. Package management for your resources is also provided by the **GLLib** class. You have access also to basic math support.
static void LZMA_Inflate ( byte[] compressDat ) throws Exception [static]
Member Data Documentation

final int Align = SpecPos + kNumFullDistances - kEndPosModelIndex [static, package]

int inputIndex [static, package]

final int IsMatch = 0 [static, package]

final int IsRep = IsMatch + (kNumStates << kNumPosBitsMax) [static, package]

final int IsRep0Long = IsRepG2 + kNumStates [static, package]

final int IsRepG0 = IsRep + kNumStates [static, package]

final int IsRepG1 = IsRepG0 + kNumStates [static, package]

final int IsRepG2 = IsRepG1 + kNumStates [static, package]

final int kAlignTableSize = (1 << kNumAlignBits) [static, package]

final int kBitModelTotal = (1 << kNumBitModelTotalBits) [static, package]

final int kEndPosModelIndex = 14 [static, package]
<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final int kLenNumHighBits</code></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td><code>final int kLenNumHighSymbols</code></td>
<td>(1 \ll kLenNumHighBits)</td>
<td></td>
</tr>
<tr>
<td><code>final int kLenNumLowBits</code></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><code>final int kLenNumLowSymbols</code></td>
<td>(1 \ll kLenNumLowBits)</td>
<td></td>
</tr>
<tr>
<td><code>final int kLenNumMidBits</code></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td><code>final int kLenNumMidSymbols</code></td>
<td>(1 \ll kLenNumMidBits)</td>
<td></td>
</tr>
<tr>
<td><code>final int kMatchMinLen</code></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td><code>final int kNumAlignBits</code></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td><code>final int kNumBitModelTotalBits</code></td>
<td>11</td>
<td></td>
</tr>
<tr>
<td><code>final int kNumFullDistances</code></td>
<td>(1 \ll (kEndPosModelIndex &gt;&gt; 1))</td>
<td></td>
</tr>
<tr>
<td><code>final int kNumLenProbs</code></td>
<td>(\text{LenHigh} + kLenNumHighSymbols)</td>
<td></td>
</tr>
<tr>
<td><code>final int kNumLenToPosStates</code></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Value</td>
<td>Modifier</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>final int <code>kNumMoveBits</code></td>
<td>5</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kNumPosBitsMax</code></td>
<td>4</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kNumPosSlotBits</code></td>
<td>6</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kNumPosStatesMax</code></td>
<td>((1 &lt;&lt; kNumPosBitsMax))</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kNumStates</code></td>
<td>12</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kNumTopBits</code></td>
<td>24</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kStartPosModelIndex</code></td>
<td>4</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>kTopValue</code></td>
<td>((1 &lt;&lt; kNumTopBits))</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>LenChoice</code></td>
<td>0</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>LenChoice2</code></td>
<td><code>LenChoice</code> + 1</td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>LenCoder</code></td>
<td><code>Align</code> + <code>kAlignTableSize</code></td>
<td>[static, package]</td>
</tr>
<tr>
<td>final int <code>LenHigh</code></td>
<td><code>LenMid</code> + (<code>kNumPosStatesMax</code> (\ll) <code>kLenNumMidBits</code>)</td>
<td>[static, package]</td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
<td>Package</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>LenLow</td>
<td>LenChoice2 + 1</td>
<td>static, package</td>
</tr>
<tr>
<td>LenMid</td>
<td>LenLow + (kNumPosStatesMax &lt;&lt; kLenNumLowBits)</td>
<td>static, package</td>
</tr>
<tr>
<td>Literal</td>
<td>RepLenCoder + kNumLenProbs</td>
<td>static, package</td>
</tr>
<tr>
<td>LZMA_BASE_SIZE</td>
<td>1846</td>
<td>static, package</td>
</tr>
<tr>
<td>LZMA_LIT_SIZE</td>
<td>768</td>
<td>static, package</td>
</tr>
<tr>
<td>LZMA_RESULT_DATA_ERROR</td>
<td>1</td>
<td>static, package</td>
</tr>
<tr>
<td>LZMA_RESULT_NOT_ENOUGH_MEM</td>
<td>2</td>
<td>static, package</td>
</tr>
<tr>
<td>LZMA_RESULT_OK</td>
<td>0</td>
<td>static, package</td>
</tr>
<tr>
<td>m_Buffer</td>
<td>byte []</td>
<td>static, package</td>
</tr>
<tr>
<td>m_Code</td>
<td>long</td>
<td>static, package</td>
</tr>
<tr>
<td>m_ExtraBytes</td>
<td>int</td>
<td>static, package</td>
</tr>
<tr>
<td>m_inSize</td>
<td>int</td>
<td>static, package</td>
</tr>
<tr>
<td>Line</td>
<td>Code</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>byte [] m_outStream [static, package]</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>long m_Range [static, package]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>final int PosSlot = IsRep0Long + (kNumStates &lt;&lt; kNumPosBitsMax) [static, package]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>final int RepLenCoder = LenCoder + kNumLenProbs [static, package]</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>final int SpecPos = PosSlot + (kNumLenToPosStates &lt;&lt; kNumPosSlotBits) [static, package]</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>final int tkNumMoveBits = 5 [static, package]</td>
<td></td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:32 2008 for GLLib by doxygen 1.5.2*
GLLibConfig Class Reference
[GLLib Configuration file]

**GLLib** setup class. [More...]

[List of all members.]
## Static Public Attributes

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static boolean</td>
<td>disableNotifyDestroyed = false</td>
<td></td>
<td>Use this option to remove the call to notifyDestroyed() at the end of the application loop i.e.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_Network_ErrorNoCredits = 4</td>
<td></td>
<td>Doja Error Messages.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_Network_ErrorNoNetwork = 0</td>
<td></td>
<td>Doja Error Messages.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_Network_ErrorNoNetworkAccess = 1</td>
<td></td>
<td>Doja Error Messages.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_Network_NoError = -1</td>
<td></td>
<td>Doja Error Messages.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_ScratchPad_CreditsFile = -1</td>
<td></td>
<td>Doja ScratchPad Indexes.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_ScratchPad_EOF = -1</td>
<td></td>
<td>Doja ScratchPad Indexes.</td>
</tr>
<tr>
<td>static int</td>
<td>doja_ScratchPad_SaveGameFile = -1</td>
<td></td>
<td>Doja ScratchPad Indexes.</td>
</tr>
</tbody>
</table>
**Warning:**
*Doja ScratchPad Indexes.*

static int **FakelInterruptThreshold** = 3000

**Warning:**
*If the interruption duration is longer than this value, then the game will know that it was interrupted.*

static int **FIXED_PRECISION** = 8

Default MIDP 1 : 8
Default MIDP 1 NOKIA : 8
Default MIDP 1 SPRINT : 8
Default MIDP 2 : 8
Default MIDP 2 NOKIA : 8
Default MIDP 2 SPRINT : 8
Default DOJA : 8
Default BLACKBERRY : 8 Default WIPI_JAVA : 8

static int **FPSLimiter** = 25

**Warning:**
*Limit the game speed to this fps value.*

static int **keycodeDown** = -2

**Warning:**
*Default Key Value Default MIDP 1 : -2
Default MIDP 1 NOKIA : -2
Default MIDP 1 SPRINT : -2
Default MIDP 2 : -2
Default MIDP 2 NOKIA : -2
Default MIDP 2 SPRINT : -2
Default DOJA : -2
Default BLACKBERRY : -2 Default WIPI_JAVA : -2*

static int **keycodeFire** = -5

**Warning:**
*Default Key Value Default MIDP 1 : -5
Default MIDP 1 NOKIA : -5
Default MIDP 1 SPRINT : -5
Default MIDP 2 : -5
Default MIDP 2 NOKIA : -5
Default MIDP 2 SPRINT : -5
Default DOJA : -5
Default BLACKBERRY : -5 Default WIPI_JAVA : -5*

static int **keycodeLeft** = -3
static int keycodeLeftSoftkey = -6

Warning:
Default Key Value Default MIDP 1 : -6
Default MIDP 1 NOKIA : -6
Default MIDP 1 SPRINT : -6
Default MIDP 2 : -6
Default MIDP 2 NOKIA : -6
Default MIDP 2 SPRINT : -6
Default DOJA : -6
Default BLACKBERRY : -6 Default WIPI_JAVA : -6

static int keycodeRight = -4

Warning:
Default Key Value Default MIDP 1 : -4
Default MIDP 1 NOKIA : -4
Default MIDP 1 SPRINT : -4
Default MIDP 2 : -4
Default MIDP 2 NOKIA : -4
Default MIDP 2 SPRINT : -4
Default DOJA : -4
Default BLACKBERRY : -4 Default WIPI_JAVA : -4

static int keycodeRightSoftkey = -7

Warning:
Default Key Value Default MIDP 1 : -7
Default MIDP 1 NOKIA : -7
Default MIDP 1 SPRINT : -7
Default MIDP 2 : -7
Default MIDP 2 NOKIA : -7
Default MIDP 2 SPRINT : -7
Default DOJA : -7
Default BLACKBERRY : -7 Default WIPI_JAVA : -7

static int keycodeUp = -1

Warning:
Default Key Value Default MIDP 1 : -1
Default MIDP 1 NOKIA : -1
<table>
<thead>
<tr>
<th>Static Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lowMemoryLimit</code></td>
<td>static int</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Low memory value (in byte).</td>
</tr>
<tr>
<td><code>math_angleFixedPointBase</code></td>
<td>static int</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Angle fixed point base (angle goes from 0 to $(1&lt;&lt;\text{Math_AngleFixedPointBase})$, $0=0$, and $(1&lt;&lt;\text{Math_AngleFixedPointBase})=360$ Default MIDP 1 : 8 Default MIDP 1 NOKIA : 8 Default MIDP 1 SPRINT : 8 Default MIDP 2 : 8 Default MIDP 2 NOKIA : 8 Default MIDP 2 SPRINT : 8 Default DOJA : 8 Default BLACKBERRY : 8 Default WIPI_JAVA : 8</td>
</tr>
<tr>
<td><code>math_AtanUseCacheTable</code></td>
<td>static boolean</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Use a table to store Atan result, greatly speed up Atan, but consume $(1&lt;&lt;\text{math_fixedPointBase})+1$ int to store table Default MIDP 1 : true Default MIDP 1 NOKIA : true Default MIDP 1 SPRINT : true Default MIDP 2 : true Default MIDP 2 NOKIA : true Default MIDP 2 SPRINT : true Default DOJA : true Default BLACKBERRY : true Default WIPI_JAVA : true</td>
</tr>
<tr>
<td><code>math_fixedPointBase</code></td>
<td>static int</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Fixed point base (default = 8, but 6 is a good candidate).</td>
</tr>
<tr>
<td><code>MAX_FLIP_COUNT</code></td>
<td>static int</td>
</tr>
<tr>
<td>Default MIDP 1 : 3 Default MIDP 1 NOKIA : 3 Default MIDP 1 SPRINT : 3 Default MIDP 2 : 3</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>MAX_MODULE_MAPPINGS</code></td>
<td>static int</td>
</tr>
<tr>
<td><code>MAX_SPRITE_PALETTEs</code></td>
<td>static int</td>
</tr>
<tr>
<td><code>MAX_WRAP_TEXT_INFO</code></td>
<td>static int</td>
</tr>
<tr>
<td><code>MIDP2forceNonFullScreen</code></td>
<td>static boolean</td>
</tr>
<tr>
<td><code>pack_dbgDataAccess</code></td>
<td>static boolean</td>
</tr>
<tr>
<td><code>pack_keepLoaded</code></td>
<td>static boolean</td>
</tr>
</tbody>
</table>

**Warning:**

- For `MAX_MODULE_MAPPINGS`: How many module mappings will be supported.
- For `MAX_SPRITE_PALETTEs`: How many palette the sprite with the largest palette count will have.
- For `MAX_WRAP_TEXT_INFO`: How many wrap text info will be supported.
- For `MIDP2forceNonFullScreen`: Force MIDP2 device to stay in non full screen mode. If set to true, the device will remain in non full screen mode.
- For `pack_dbgDataAccess` and `pack_keepLoaded`: File/package debug trigger and once a pack or subpack is loaded, keep it there respectively.
<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td><strong>pack_skipbufferSize</strong> = 256</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>When InputStream.skip() is slower than a read(), we use a skip buffer by reading the data to skip/discard into a temporary buffer.</em></td>
</tr>
<tr>
<td>static boolean</td>
<td><strong>pack_supportLZMADecompression</strong> = false</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>Support LZMA decompression for resources and streams.</em></td>
</tr>
<tr>
<td>static boolean</td>
<td><strong>pack_useBlackBerryGZipDecompression</strong> = false</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>(BlackBerry only) If data is compressed with Gzip (to reduce .cod size), enable this flag.</em></td>
</tr>
<tr>
<td>static boolean</td>
<td><strong>pathfinding_Debug</strong> = false</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>Use this option to see all the pathfinding information in the console.</em></td>
</tr>
<tr>
<td>static int</td>
<td><strong>pathfinding_MaxNode</strong> = 400</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>Maximum node count the algorithm could visit when trying to find a path.</em></td>
</tr>
<tr>
<td>static boolean</td>
<td><strong>platformRequestOnExit</strong> = true</td>
</tr>
<tr>
<td>Warning:</td>
<td><em>This boolean indicates if platformRequest should be done when exiting the app.</em></td>
</tr>
<tr>
<td>static int</td>
<td><strong>PNG_BUFFER_SIZE</strong> = 256</td>
</tr>
<tr>
<td>Default MIDP 1 :</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA :</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT :</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 2 :</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA :</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT :</td>
<td>256</td>
</tr>
<tr>
<td>Default DOJA :</td>
<td>256</td>
</tr>
<tr>
<td>Default BLACKBERRY :</td>
<td>256</td>
</tr>
<tr>
<td>Default WIPI JAVA :</td>
<td>20 * 1024</td>
</tr>
<tr>
<td>Variable Type</td>
<td>Variable Name</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>rms_usePackRead</code></td>
</tr>
<tr>
<td>static boolean</td>
<td><code>rms_useSharing</code></td>
</tr>
<tr>
<td>static int</td>
<td><code>screenHeight</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td><code>screenWidth</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>static int</td>
<td><code>SLEEP_DRAWSTRINGB</code></td>
</tr>
<tr>
<td>static boolean</td>
<td><code>softkeyOKOnLeft</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><code>sound_enable</code></td>
<td><code>true</code></td>
</tr>
<tr>
<td><code>sound_enableThread</code></td>
<td><code>true</code></td>
</tr>
<tr>
<td><code>sound_numberOfChannels</code></td>
<td><code>2</code></td>
</tr>
<tr>
<td><code>sound_useCachedPlayers</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td><code>sound_useFakeMediaDuration</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td><code>sound_useFreeChannelOnStop</code></td>
<td><code>false</code></td>
</tr>
<tr>
<td><code>sound_useJSR135</code></td>
<td><code>true</code></td>
</tr>
<tr>
<td><code>sound_usePrefetchedPlayers</code></td>
<td><code>false</code></td>
</tr>
</tbody>
</table>
### Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static boolean sound_useRealizedPlayers</td>
<td>false</td>
<td><em>Warning:</em> Set all sound players to prefetched state at loading time.</td>
</tr>
<tr>
<td>static boolean sound_useSetLevel</td>
<td>true</td>
<td><em>Warning:</em> Enables using set level for control the volume of players.</td>
</tr>
<tr>
<td>static boolean sound_useSetMediaTimeBeforePlay</td>
<td>true</td>
<td><em>Warning:</em> Triggers Player.setMediaTime(0) before sound play.</td>
</tr>
<tr>
<td>static boolean sound_useStopSoundsOnInterrupt</td>
<td>true</td>
<td><em>Warning:</em> Forces to stop sounds upon interrupt.</td>
</tr>
<tr>
<td>static boolean sprite_alwaysBsNfm1Byte</td>
<td>false</td>
<td><em>Warning:</em> Use if Aurora exported the number of frame modules as 1 byte.</td>
</tr>
<tr>
<td>static boolean sprite_alwaysBsNoAfStart</td>
<td>false</td>
<td><em>Warning:</em> Use if Aurora did not export start offsets for frames, they are one after the other.</td>
</tr>
<tr>
<td>static boolean sprite_alwaysBsNoFmStart</td>
<td>false</td>
<td><em>Warning:</em> Use if Aurora did not export start offsets for frame modules, they are one after the other.</td>
</tr>
<tr>
<td>static boolean sprite_alwaysBsSkipFrameRc</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Variable Type</td>
<td>Variable Name</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>static int</td>
<td><code>sprite_animFPS</code></td>
<td>= 25</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_debugColision</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_debugErrors</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_debugLoading</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_debugModuleUsage</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_debugUsedMemory</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_drawPixelClippingBug</code></td>
<td>= false</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_drawRegionFlippedBug</code></td>
<td>= false</td>
</tr>
<tr>
<td>Warning:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If your device has a bugged <code>drawRegion</code> with flipped, this function will be replaced by <code>drawImage</code>.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```java
static boolean sprite_drawRGBTransparencyBug = false
```

**Warning:**

- Only use this flag when `drawRGB` can not draw transparency (Ex: Softbank's Samsung MIDP2 phones)
- Default MIDP 1: false // INVALID
- Default MIDP 1 NOKIA: false // INVALID
- Default MIDP 1 SPRINT: false // INVALID
- Default MIDP 2: false
- Default MIDP 2 NOKIA: false
- Default MIDP 2 SPRINT: false
- Default DOJA: false // INVALID
- Default BLACKBERRY: false // INVALID
- Default WIPI_JAVA: false // INVALID

```java
static boolean sprite_fillRoundRectBug = false
```

**Warning:**

If your device has a bugged `fillRoundRect`, internal call to this function will be replaced by `fillRect`.

```java
static boolean sprite_fontBackslashChangePalette = false
```

**Warning:**

When drawing a string, the `\` char is used to changed palette, it has to be followed by a number corresponding to the palette number.

```java
static boolean sprite_fontUseOneFramePerLetter = false
```

**Warning:**

Each Letter Of The Font Is Mapped To Its Own Frame.

```java
static boolean sprite_fpsRegion = false
```

**Warning:**

Don't use clipping when drawing sprites with `drawRegion` by manipulating the call's parameters.

```java
static boolean sprite_ModuleMapping_useModuleImages = true
```

**Warning:**
Each module will be exported as its own images.

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_newTextRendering = true</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Use new text rendering code Default MIDP 1 : true</td>
</tr>
<tr>
<td></td>
<td>Default MIDP 1 NOKIA : true</td>
</tr>
<tr>
<td></td>
<td>Default MIDP 1 SPRINT : true</td>
</tr>
<tr>
<td></td>
<td>Default MIDP 2 : true</td>
</tr>
<tr>
<td></td>
<td>Default MIDP 2 NOKIA : true</td>
</tr>
<tr>
<td></td>
<td>Default MIDP 2 SPRINT : true</td>
</tr>
<tr>
<td></td>
<td>Default DOJA : true</td>
</tr>
<tr>
<td></td>
<td>Default BLACKBERRY : true</td>
</tr>
<tr>
<td></td>
<td>Default WIPI_JAVA : true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_RGBArraysUseDrawRGB = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>When using sprite_ModuleMapping_useModuleImages and NOT using sprite_useCacheRGBArrays, transform images directly on the image’s data in int[].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_useAfOffShort = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Use if Aurora exported animation frames offsets as short[] instead of byte[].</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_useBSpriteFlags = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Use Bsprite Flags.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_useBugFixImageOddSize = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Some phone don’t support the creation of Image with Odd width.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th>sprite_useCacheFlipXY = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Keep a version of the Single Image flipped in X and another flipped in Y.</td>
</tr>
</tbody>
</table>

<p>| static boolean | sprite_useCachePool = false |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static boolean sprite_useCacheRGBArrays</td>
<td>= false</td>
<td>Keep the sprite's images to cache pool.</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>Keep the sprite's images as int[].</td>
</tr>
<tr>
<td>static boolean sprite_useCreateRGB</td>
<td>= true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>The sprite's Images are in RAW data format in the resources.</td>
</tr>
<tr>
<td>static boolean sprite_useCreateRGBTransparencyBug</td>
<td>= false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>On some devices Image.createRGBImage the bProccessAlpha is bugged and must be always true, if tis your case, use this option.</td>
</tr>
<tr>
<td>static boolean sprite_useDeactivateSystemGc</td>
<td>= false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>Use this option to disable calls to System.gc() everywhere in the sprite management.</td>
</tr>
<tr>
<td>static boolean sprite_useDrawRegionClipping</td>
<td>= false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>Simulate drawregion by using drawImage and Intersection of setClip.</td>
</tr>
<tr>
<td>static boolean sprite_useDrawStringSleep</td>
<td>= false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>On some low end device drawing large string on screen will choke the pipeline because there are too many drawImages, some phone will get a speed increase by doing tiny sleep at the end of each draw strings.</td>
</tr>
<tr>
<td>static boolean sprite_useDynamicPng</td>
<td>= false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
<td>The sprite's Images are Dechuncked PNG in the resources.</td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat127RLE = true</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 127 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat16 = true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 16 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat2 = true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 2 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat256 = false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 256 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat256RLE = true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 256 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat4 = true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 4 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useEncodeFormat64RLE = true</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Support for Image containg 64 colors or less.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useExternImage = false</td>
<td></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>If you have special way to create, load, access the module's images.</td>
<td></td>
</tr>
<tr>
<td>static boolean</td>
<td>sprite_useFMOffShort = false</td>
<td></td>
</tr>
</tbody>
</table>
### Warning:
Keep internal position of a module Inside a frame in short[]. instead of byte[].

<table>
<thead>
<tr>
<th>Warning:</th>
<th>Use if Aurora exported the Palette to use for each module in a frame (fmodule).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static boolean</strong></td>
<td><code>sprite_useFMPalette</code> = false</td>
</tr>
</tbody>
</table>

### Warning:
Your sprite’s frames are using collision box.

<table>
<thead>
<tr>
<th>Warning:</th>
<th>If you have exported your sprite with frame rect, you have to load them.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static boolean</strong></td>
<td><code>sprite_useFrameCollRC</code> = false</td>
</tr>
</tbody>
</table>

### Warning:
If the sprite have more than 256 Frames per sprite you can extend to

<table>
<thead>
<tr>
<th>Warning:</th>
<th>Use if your sprite contains some Hyperframe.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static boolean</strong></td>
<td><code>sprite_useHyperFM</code> = false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Warning:</th>
<th>GenPalette generates a palette based on another palette.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static int</strong></td>
<td><code>sprite_useGenPalette</code> = 1 &lt;&lt; ASprite.PAL_GREY</td>
</tr>
</tbody>
</table>

### Warning:
Use if your sprite contains some Hyperframe.

<table>
<thead>
<tr>
<th>Warning:</th>
<th>If the sprite have more than 256 Frames per sprite you can extend to</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static boolean</strong></td>
<td><code>sprite_useIndexExAframes</code> = false</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>sprite_useIndexExFmodules</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useLoadImageWithoutTransf</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleColorAsByte</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleDataOffAsShort</td>
<td>true</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleMapping</td>
<td>true</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleUsageFromSprite</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleWHShort</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>sprite_useModuleXY</td>
<td>false</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>Variable Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>sprite_useModuleXYShort</code></td>
<td>Use/Keep each module position in a Single Image.</td>
</tr>
<tr>
<td><code>sprite_useMultipleModuleTypes</code></td>
<td>Use for recording drawing operations.</td>
</tr>
<tr>
<td><code>sprite_useNokia7650DrawPixelBug</code></td>
<td>For really Old Nokia firmware, some vertical flip where not working, this option will flip the data manually.</td>
</tr>
<tr>
<td><code>sprite_useNokiaUI</code></td>
<td>Do you want to use NokiaUI API.</td>
</tr>
<tr>
<td><code>sprite_useNonInterlaced</code></td>
<td>Use this flag to load the Sprite exported by Aurora with .fft config file.</td>
</tr>
<tr>
<td><code>sprite_useOperationMark</code></td>
<td>Use for recording drawing operations.</td>
</tr>
<tr>
<td><code>sprite_useOperationRecord</code></td>
<td>Use for recording drawing operations.</td>
</tr>
<tr>
<td>Static boolean</td>
<td>Property</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td><code>sprite_useOperationRect</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use for recording drawing operations.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePixelFormat0565</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0565 Palette color encoding.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePixelFormat1555</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1555 Palette color encoding.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePixelFormat4444</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4444 Palette color encoding.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePixelFormat8888</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8888 Palette color encoding.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePrecomputedCRC</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Are the CRC exported with the image’s data? For RGB or Nokia’s phone, don’t use this.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_usePrecomputedFrameRect</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Frame’s Position and Size are exported in the sprite data at compile time.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>sprite_useResize</code></td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Support sprite/module resize.</td>
</tr>
<tr>
<td>static boolean</td>
<td><code>sprite_useSingleArrayForFMAF</code> = false</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Internal use of a single array for the module’s informations and a single array for the frame’s informations.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useSingleDirectGraphics</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>On some Nokia devices getting the DirectGraphics can be slow or create memory leaks.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useSingleImageForAllModules</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>If your device can’t load a lot (count, not size) of images, this option will use only one big image for all the modules.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useSkipFastVisibilityTest</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>In PaintModule, check to see if the module is in part in the current clipping selection.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useTransfFlip</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>When drawing a flipped Frame/Module, apply a translation to keep Frame/Module Cohesion.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useTransfRot</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>When drawing a Rotated Frame/Module, apply a translation to keep Frame/Module Cohesion.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static boolean</th>
<th><code>sprite_useTruncatedRGBBuffer</code> = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Modify the <code>sprite_RGBArraysUseDrawRGB</code> option by drawing the image without going through a temporary Image.</td>
</tr>
<tr>
<td>Static boolean</td>
<td>Text Use Internal UTF8 Converter = false</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td><strong>Warning:</strong></td>
<td>Used to go around some firmware bug of UTF-8 byte[] to String convertions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Static boolean</th>
<th>Text Use String Cache = true</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>If true text are stored in String (consume more memory, but less allocation), if false, text are stored as byte array (less memory) and converted each time a String is needed. Default MIDP 1 : true</td>
</tr>
</tbody>
</table>

| Default MIDP 1 | NOKIA : true |
| Default MIDP 1 | SPRINT : true |
| Default MIDP 2 | true |
| Default MIDP 2 | NOKIA : true |
| Default MIDP 2 | SPRINT : true |
| Default DOJA   | true |
| Default BLACKBERRY | true |
| Default WIPI JAVA | true |

<table>
<thead>
<tr>
<th>Static int</th>
<th>Tileset Max Layer Count = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>How many layer the tileset with the largest layer count will have.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Static boolean</th>
<th>Tileset Use Index As Short = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Use short instead of byte to keep tileset indexes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Static boolean</th>
<th>Tileset Use Tile Shift = true</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td>Set to true if you work with power of 2 tile size (recommended).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Static int</th>
<th>TMP Buffer Size = 256</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default MIDP 1</strong></td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 1</td>
<td>NOKIA : 256</td>
</tr>
<tr>
<td>Default MIDP 1</td>
<td>SPRINT : 256</td>
</tr>
<tr>
<td>Default MIDP 2</td>
<td>256</td>
</tr>
<tr>
<td>Default MIDP 2</td>
<td>NOKIA : 256</td>
</tr>
<tr>
<td>Default MIDP 2</td>
<td>SPRINT : 256</td>
</tr>
<tr>
<td>Default DOJA</td>
<td>256</td>
</tr>
<tr>
<td>Default BLACKBERRY</td>
<td>256</td>
</tr>
<tr>
<td>Default WIPI JAVA</td>
<td>256</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Static boolean</th>
<th>Use Absolute Value Of KeyCode = false</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>Static boolean</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>useBugFixMultipleKeyPressed = false</td>
<td>Warning: Fix the bug that some keys are not released when multiple keys are pressed.</td>
</tr>
<tr>
<td>useCallSerially = false</td>
<td>Warning: May solve some slowness/key response.</td>
</tr>
<tr>
<td>useDrawLineClippingBug = false</td>
<td>Warning: Fix for the drawLine method bug, when it draws outside the current clip if $y_1 &gt; y_2$.</td>
</tr>
<tr>
<td>useDrawPartialRGB = false</td>
<td>Warning: For devices that drawing with RGB outside the screen make them crash.</td>
</tr>
<tr>
<td>useFakeInterruptHandling = false</td>
<td>Warning: For devices which don’t catch interrupt properly (i.e.</td>
</tr>
<tr>
<td>useFlashLightInsteadOfVibration = false</td>
<td></td>
</tr>
</tbody>
</table>

Use absolute value for keycode, if keycode can be positive or negative (as to maintain compatibility on triplets phones for example) Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false  Default WIPI_JAVA : false
### Warning:
If set to true, flash backLight instead of using vibration when calling Vibrate(int duration) function

- Default MIDP 1: false // INVALID
- Default MIDP 1 NOKIA: false // INVALID
- Default MIDP 1 SPRINT: false // INVALID
- Default MIDP 2: false // Warning, be careful
- Default MIDP 2 NOKIA: false // Warning, be careful
- Default MIDP 2 SPRINT: false // Warning, be careful
- Default DOJA: false // Warning, be careful
- Default BLACKBERRY: false // INVALID
- Default WIPI_JAVA: false // INVALID

---

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static boolean useFrameDT</strong></td>
<td>true</td>
<td>Use DeltaTime between each frame.</td>
</tr>
<tr>
<td><strong>static boolean useKeyAccumulation</strong></td>
<td>true</td>
<td>Key accumulation option.</td>
</tr>
<tr>
<td><strong>static boolean useSafeDrawRegion</strong></td>
<td>true</td>
<td>May prevent exceptions and slowdowns during drawing.</td>
</tr>
<tr>
<td><strong>static boolean useSafeFillRect</strong></td>
<td>false</td>
<td>Workarounds the problem of fillRect() having no effect on a backbuffer outside the size of the screen.</td>
</tr>
<tr>
<td><strong>static boolean useServiceRepaints</strong></td>
<td>true</td>
<td>May solve some devices will freeze if calling serviceRepaints.</td>
</tr>
<tr>
<td><strong>static boolean useSleepInsteadOfYield</strong></td>
<td>true</td>
<td>May solve slowness if the game looks like it's using too much CPU</td>
</tr>
</tbody>
</table>
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

static boolean useSoftwareDoubleBuffer = false

**Warning:**
Set to true to enable a software double buffer Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false Default WIPI_JAVA : false

static boolean useSystemGc = true

**Warning:**
System.gc() will be called automatically (with GLib.Gc()) if some resource be destroyed in GLib internal, set to false to deactivate this feature, and do it yourselfly.

static boolean xplayer_CARRIER_MXTELCEL = false

**Warning:**
Set to true if the game is intended for MX Telcel.

static boolean xplayer_CARRIER_USCINGULAR_BLUE = false

**Warning:**
Set to true if the game is intended for US Cingular Blue network.

static boolean xplayer_CARRIER_USCINGULAR_ORANGE = false

**Warning:**
Set to true if the game is intended for US Cingular Orange network.

static boolean xplayer_CARRIER_USNEXTEL = false

**Warning:**
Set to true if the game is intended for US NEXTEL.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>xplayer_CARRIER_USSPRINT</code></td>
<td>boolean, Set to true if the game is intended for US Sprint.</td>
</tr>
<tr>
<td><code>xplayer_CARRIER_USVIRGIN</code></td>
<td>boolean, Set to true if the game is intended for US Virgin Mobile.</td>
</tr>
<tr>
<td><code>xplayer_CONN_TIMEOUT</code></td>
<td>int, 30000, The time in milliseconds after which a server request is timed out if a response is not received.</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_DEBUG</code></td>
<td>boolean, true, Enables debug print statements.</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_DUAL_TCP</code></td>
<td>boolean, false, Enables 2 different connections for sending and receiving data using TCP.</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_FIND_PLAYER</code></td>
<td>boolean, false, Enables the search player feature.</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_M7_SUPPORT</code></td>
<td>boolean, false, Auto defined for M7 operators.</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_MULTIPLAYER</code></td>
<td>boolean, false, Enables support for multiplayer games using TCP connections.</td>
</tr>
<tr>
<td>Variable</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_PLAYER_SPECIFIC_DATA</code></td>
<td>false</td>
</tr>
<tr>
<td><code>xplayer_ENABLE_TIMEOUT</code></td>
<td>false</td>
</tr>
<tr>
<td><code>xplayer_HTTP_NO_CANCEL</code></td>
<td>false</td>
</tr>
<tr>
<td><code>xplayer_KEEP_ALIVE_TIME</code></td>
<td>7000</td>
</tr>
<tr>
<td><code>xplayer_USE_BUG_FIX_MESSAGE_SIZE</code></td>
<td>false</td>
</tr>
<tr>
<td><code>xplayer_USE_HTTP_POST</code></td>
<td>false</td>
</tr>
<tr>
<td><code>xplayer_XPLAYER_VERSION</code></td>
<td>1</td>
</tr>
</tbody>
</table>
Detailed Description

**GLLib** setup class.

This class holds all the constants and option to work with the **GLLib**. In release all of these value will be final, so you should not modify those, they should be optimized for your device.
## Member Data Documentation

### boolean disableNotifyDestroyed = false [static]

**Warning:**

Use this option to remove the call to notifyDestroyed() at the end of the application loop i.e.

When your application is quitting Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false // INVALID
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false

### int doja_Network_ErrorNoCredits = 4 [static]

**Warning:**

Doja Error Messages.

Default MIDP 1 : 4
Default MIDP 1 NOKIA : 4
Default MIDP 1 SPRINT : 4
Default MIDP 2 : 4
Default MIDP 2 NOKIA : 4
Default MIDP 2 SPRINT : 4
Default DOJA : 2
Default BLACKBERRY : 4 Default WIPI_JAVA : 4

### int doja_Network_ErrorNoNetwork = 0 [static]

**Warning:**

Doja Error Messages.

Default MIDP 1 : 0
Default MIDP 1 NOKIA : 0
Default MIDP 1 SPRINT : 0
Warning:
Doja Error Messages.

Default MIDP 1 : 1
Default MIDP 1 NOKIA : 1
Default MIDP 1 SPRINT : 1
Default MIDP 2 : 1
Default MIDP 2 NOKIA : 1
Default MIDP 2 SPRINT : 1
Default DOJA : 1
Default BLACKBERRY : 1 Default WIPI_JAVA : 1

Warning:
Doja Error Messages.

Default MIDP 1 : -1
Default MIDP 1 NOKIA : -1
Default MIDP 1 SPRINT : -1
Default MIDP 2 : -1
Default MIDP 2 NOKIA : -1
Default MIDP 2 SPRINT : -1
Default DOJA : -1
Default BLACKBERRY : -1 Default WIPI_JAVA : -1

Warning:
Doja ScratchPad Indexes.

Default MIDP 1 : -1
Warning:

Doja ScratchPad Indexes.

Default MIDP 1 NOKIA : -1
Default MIDP 1 SPRINT : -1
Default MIDP 2 : -1
Default MIDP 2 NOKIA : -1
Default MIDP 2 SPRINT : -1
Default DOJA : -1
Default BLACKBERRY : -1 Default WIPI_JAVA : -1

------------------

Warning:

Doja ScratchPad Indexes.

Default MIDP 1 : -1
Default MIDP 1 NOKIA : -1
Default MIDP 1 SPRINT : -1
Default MIDP 2 : -1
Default MIDP 2 NOKIA : -1
Default MIDP 2 SPRINT : -1
Default DOJA : -1
Default BLACKBERRY : -1 Default WIPI_JAVA : -1

------------------

Warning:

Doja ScratchPad Indexes.

Default MIDP 1 : -1
Default MIDP 1 NOKIA : -1
Default MIDP 1 SPRINT : -1
Default MIDP 2 : -1
Default MIDP 2 NOKIA : -1
Default MIDP 2 SPRINT : -1
Default DOJA : -1
Default BLACKBERRY : -1 Default WIPI_JAVA : -1

------------------

Warning:

If the interruption duration is longer than this value, then the game
will know that it was interrupted.

Default value: 3000 (milliseconds). If there's issue like "in-game menu unexpectedly opens after loading from main-menu to in-game", try increasing this value.

Default MIDP 1 : 3000
Default MIDP 1 NOKIA : 3000
Default MIDP 1 SPRINT : 3000
Default MIDP 2 : 3000
Default MIDP 2 NOKIA : 3000
Default MIDP 2 SPRINT : 3000
Default DOJA : 3000
Default BLACKBERRY : 3000 Default WIPI_JAVA : 3000

```java
int FIXED_PRECISION = 8 [static]
```

Default MIDP 1 : 8
Default MIDP 1 NOKIA : 8
Default MIDP 1 SPRINT : 8
Default MIDP 2 : 8
Default MIDP 2 NOKIA : 8
Default MIDP 2 SPRINT : 8
Default DOJA : 8
Default BLACKBERRY : 8 Default WIPI_JAVA : 8

```java
int FPSlimiter = 25 [static]
```

**Warning:**

Limit the game speed to this fps value.

Default MIDP 1 : 25
Default MIDP 1 NOKIA : 25
Default MIDP 1 SPRINT : 25
Default MIDP 2 : 25
Default MIDP 2 NOKIA : 25
Default MIDP 2 SPRINT : 25
Default DOJA : 25
Default BLACKBERRY : 25 Default WIPI_JAVA : 25

```java
int keycodeDown = -2 [static]
```
Warning:

Default Key Value Default MIDP 1 : -2
Default MIDP 1 NOKIA : -2
Default MIDP 1 SPRINT : -2
Default MIDP 2 : -2
Default MIDP 2 NOKIA : -2
Default MIDP 2 SPRINT : -2
Default DOJA : -2
Default BLACKBERRY : -2 Default WIPI_JAVA : -2

int keycodeFire = -5 [static]

Warning:

Default Key Value Default MIDP 1 : -5
Default MIDP 1 NOKIA : -5
Default MIDP 1 SPRINT : -5
Default MIDP 2 : -5
Default MIDP 2 NOKIA : -5
Default MIDP 2 SPRINT : -5
Default DOJA : -5
Default BLACKBERRY : -5 Default WIPI_JAVA : -5

int keycodeLeft = -3 [static]

Warning:

Default Key Value Default MIDP 1 : -3
Default MIDP 1 NOKIA : -3
Default MIDP 1 SPRINT : -3
Default MIDP 2 : -3
Default MIDP 2 NOKIA : -3
Default MIDP 2 SPRINT : -3
Default DOJA : -3
Default BLACKBERRY : -3 Default WIPI_JAVA : -3

int keycodeLeftSoftkey = -6 [static]

Warning:

Default Key Value Default MIDP 1 : -6
Default MIDP 1 NOKIA : -6
Default MIDP 1 SPRINT : -6
Default MIDP 2 : -6
Default MIDP 2 NOKIA : -6
Default MIDP 2 SPRINT : -6
Default DOJA : -6
Default BLACKBERRY : -6
Default WIPI_JAVA : -6

int keycodeRight = -4 [static]

Warning:
Default Key Value Default MIDP 1 : -4
Default MIDP 1 NOKIA : -4
Default MIDP 1 SPRINT : -4
Default MIDP 2 : -4
Default MIDP 2 NOKIA : -4
Default MIDP 2 SPRINT : -4
Default DOJA : -4
Default BLACKBERRY : -4
Default WIPI_JAVA : -4

int keycodeRightSoftkey = -7 [static]

Warning:
Default Key Value Default MIDP 1 : -7
Default MIDP 1 NOKIA : -7
Default MIDP 1 SPRINT : -7
Default MIDP 2 : -7
Default MIDP 2 NOKIA : -7
Default MIDP 2 SPRINT : -7
Default DOJA : -7
Default BLACKBERRY : -7
Default WIPI_JAVA : -7

int keycodeUp = -1 [static]

Warning:
Default Key Value Default MIDP 1 : -1
Default MIDP 1 NOKIA : -1
Default MIDP 1 SPRINT : -1
Default MIDP 2 : -1
Default MIDP 2 NOKIA : -1
Default MIDP 2 SPRINT : -1
Default DOJA : -1
Default BLACKBERRY : -1 Default WIPI_JAVA : -1

**int** lowMemoryLimit = 0 [static]

**Warning:**
Low memory value (in byte).

System.gc() will be called automatically if the available free memory is lower than this value, set to 0 to deactivate this feature Default MIDP 1 : 0
Default MIDP 1 NOKIA : 0
Default MIDP 1 SPRINT : 0
Default MIDP 2 : 0
Default MIDP 2 NOKIA : 0
Default MIDP 2 SPRINT : 0
Default DOJA : 0
Default BLACKBERRY : 0 Default WIPI_JAVA : 0

**int** math_angleFixedPointBase = 8 [static]

**Warning:**
Angle fixed point base (angle goes from 0 to (1<<Math_AngleFixedPointBase), 0=0, and (1<<Math_AngleFixedPointBase)=360 Default MIDP 1 : 8
Default MIDP 1 NOKIA : 8
Default MIDP 1 SPRINT : 8
Default MIDP 2 : 8
Default MIDP 2 NOKIA : 8
Default MIDP 2 SPRINT : 8
Default DOJA : 8
Default BLACKBERRY : 8 Default WIPI_JAVA : 8

**boolean** math_AtanUseCacheTable = true [static]

**Warning:**
Use a table to store Atan result, greatly speed up Atan, but consume (1<<math_fixedPointBase)+1 int to store table Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

int math_fixedPointBase = 8 [static]

Warning:
Fixed point base (default = 8, but 6 is a good candidate).

Default MIDP 1 : 8
Default MIDP 1 NOKIA : 8
Default MIDP 1 SPRINT : 8
Default MIDP 2 : 8
Default MIDP 2 NOKIA : 8
Default MIDP 2 SPRINT : 8
Default DOJA : 8
Default BLACKBERRY : 8 Default WIPI_JAVA : 8

int MAX_FLIP_COUNT = 3 [static]

Default MIDP 1 : 3
Default MIDP 1 NOKIA : 3
Default MIDP 1 SPRINT : 3
Default MIDP 2 : 3
Default MIDP 2 NOKIA : 3
Default MIDP 2 SPRINT : 3
Default DOJA : 8
Default BLACKBERRY : 3 Default WIPI_JAVA : 3

int MAX_MODULE_MAPPINGS = 16 [static]

Default MIDP 1 : 16
Default MIDP 1 NOKIA : 16
Default MIDP 1 SPRINT : 16
Default MIDP 2 : 16
Default MIDP 2 NOKIA : 16
Default MIDP 2 SPRINT : 16
**Warning:**

How many palette the sprite with the largest palette count will have.

**Warning:**

Force MIDP2 device to stay in non fullscreen mode.
boolean pack_dbgDataAccess = false [static]

**Warning:**
File/Package debug trigger.

Set this variable to true to debug pack loading. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

---

boolean pack_keepLoaded = false [static]

**Warning:**
Once a pack or a subpack is loaded into memory, keep it there.

It will never get freed. Usually useful for some Nokia S60 phone which
leaks allot when loading a file. Default MIDP 1 : false // Warning, be
careful
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false // Warning, be careful
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false // Warning, be careful Default
WIPI_JAVA : false // Warning, be careful

---

int pack_skipbufferSize = 256 [static]

**Warning:**
When InputStream.skip() is slower than a read(), we use a skip
buffer by reading the data to skip/discard into a temporary buffer.

This buffer as to be as large as the largest chuck to skip in your game. Set to 0 if no skip buffer should be used. Default MIDP 1 : 256
Default MIDP 1 NOKIA : 256
Default MIDP 1 SPRINT : 256
Default MIDP 2 : 256
Default MIDP 2 NOKIA : 256
Default MIDP 2 SPRINT : 256
Default DOJA : 256
Default BLACKBERRY : 256 Default WIPI_JAVA : 256

---

boolean **pack_supportLZMADecompression** = false [static]

**Warning:**
Support LZMA decompression for resources and streams.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false // INVALID
Default BLACKBERRY : false Default WIPI_JAVA : false

---

boolean **pack_useBlackBerryGZipDecompression** = false [static]

**Warning:**
(BlackBerry only) If data is compressed with Gzip (to reduce .cod size), enable this flag.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false // INVALID
Default MIDP 2 NOKIA : false // INVALID
Default MIDP 2 SPRINT : false // INVALID
Default DOJA : false // INVALID
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID
**boolean pathfinding_Debug = false** [static]

**Warning:**
Use this option to see all the pathfinding information in the console.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

**int pathfinding_MaxNode = 400** [static]

**Warning:**
Maximum node count the algorythm could visit when trying to find a path.

Default MIDP 1 : 400
Default MIDP 1 NOKIA : 400
Default MIDP 1 SPRINT : 400
Default MIDP 2 : 400
Default MIDP 2 NOKIA : 400
Default MIDP 2 SPRINT : 400
Default DOJA : 400
Default BLACKBERRY : 400 Default WIPI_JAVA : 400

**boolean platformRequestOnExit = true** [static]

**Warning:**
This boolean indicates if platformRequest should be done when exiting the app.

Note that the responsibility of calling PlatformRequest when exiting still lays on the game. The actual call is not automatic. Default MIDP 1 : true
Default MIDP 1 NOKIA : true
### Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : false Default WIPI_JAVA : true

#### int PNG_BUFFER_SIZE = 256 [static]

Default MIDP 1 : 256
Default MIDP 1 NOKIA : 256
Default MIDP 1 SPRINT : 256
Default MIDP 2 : 256
Default MIDP 2 NOKIA : 256
Default MIDP 2 SPRINT : 256
Default DOJA : 256
Default BLACKBERRY : 256 Default WIPI_JAVA : 20 * 1024

#### boolean rms_usePackRead = false [static]

**Warning:**
Set to true if you want to use the pack filesystem when reading from the RMS.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

#### boolean rms_useSharing = false [static]

**Warning:**
Set to true if you want to read/write recordstores owned by other midlets.

Default MIDP 1 : false // INVALID
### screenHeight

**Warning:**

| Default MIDP 1 NOKIA | NOKIA | 0 |
| Default MIDP 1 SPRINT | SPRINT | 0 |
| Default MIDP 2 | 0 |
| Default MIDP 2 NOKIA | NOKIA | 0 |
| Default MIDP 2 SPRINT | SPRINT | 0 |
| Default DOJA | 0 |
| Default BLACKBERRY | BLACKBERRY | 0 |
| Default WIPI_JAVA | WIPI_JAVA | 0 |

### screenWidth

**Warning:**

| Default MIDP 1 NOKIA | NOKIA | 0 |
| Default MIDP 1 SPRINT | SPRINT | 0 |
| Default MIDP 2 | 0 |
| Default MIDP 2 NOKIA | NOKIA | 0 |
| Default MIDP 2 SPRINT | SPRINT | 0 |
| Default DOJA | 0 |
| Default BLACKBERRY | BLACKBERRY | 0 |
| Default WIPI_JAVA | WIPI_JAVA | 0 |

### SLEEP_DRAWSTRINGB

**Warning:**

| Default MIDP 1 | 1 |
| Default MIDP 1 NOKIA | NOKIA | 1 |
| Default MIDP 1 SPRINT | SPRINT | 1 |
boolean softkeyOKOnLeft = true [static]

Warning:
Softkey order Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sound_enable = true [static]

Warning:
Enable sound engine.
Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sound_enableThread = true [static]

Warning:
Use a thread to control the sound engine.
Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : false
Default BLACKBERRY : true
Default WIPI_JAVA : true

int sound_numberOfChannels = 2 [static]

**Warning:**

Number of sound channels (number of player that can run concurrently).

Default MIDP 1 : 1
Default MIDP 1 NOKIA : 1
Default MIDP 1 SPRINT : 1
Default MIDP 2 : 2
Default MIDP 2 NOKIA : 2
Default MIDP 2 SPRINT : 1
Default DOJA : 1
Default BLACKBERRY : 1
Default WIPI_JAVA : 1

boolean sound_useCachedPlayers = false [static]

**Warning:**

Cache sound players at loading time.

Using this option can increase heap memory, but can reduce lags before sound play on some phones. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false
Default WIPI_JAVA : false // INVALID

boolean sound_useFakeMediaDuration = false [static]
Warning:
Set to true to use manually provided sound durations for updating the channels states.

Use this option if Player.getState() doesn't work properly on the target device. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sound_useFreeChannelOnStop = false [static]

Warning:
Set this to true if you want to free the channel on sound stop.

Use this option if sounds dissapear after some time (for example Motorola Triples). Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sound/useJSR135 = true [static]

Warning:
Use JSR135 extension.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : false
boolean sound_usePrefetchedPlayers = false [static]

Warning:
Set all sound players to prefetched state at loading time.

Using this option can prevent/reduce lags before sound play on some phones. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sound_useRealizedPlayers = false [static]

Warning:
Set all sound players to realized state at loading time.

Using this option can prevent/reduce lags before sound play on some phones. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sound_useSetLevel = true [static]

Warning:
Enables using set level for control the volume of players.

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sound_useSetMediaTimeBeforePlay = true [static]

Warning:
Triggers Player.setMediaTime(0) before sound play.

Using this option forces to start the sound to play from beginning, but can lead to sound glitches (mainly for devices which do not support the pause/resume option). Default MIDP 1 : true // INVALID
Default MIDP 1 NOKIA : true // INVALID
Default MIDP 1 SPRINT : true // INVALID
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true // INVALID

boolean sound_useStopSoundsOnInterrupt = true [static]

Warning:
Forces to stop sounds upon interrupt.

Set this option to false to if the jvm automatically stops sounds upon interrupt, and sounds are lost after that. (Reference: https://wiki.gameloft.org/twiki/bin/view/Main/MotorolaTriplets#Interrupts
Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true
boolean sprite_alwaysBsNfm1Byte = false [static]

Warning:
Use if Aurora exported the number of frame modules as 1 byte.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_alwaysBsNoAfStart = false [static]

Warning:
Use if Aurora did not export start offsets for frames, they are one after the other.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_alwaysBsNoFmStart = false [static]

Warning:
Use if Aurora did not export start offsets for frame modules, they are one after the other.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
### boolean `sprite_alwaysBSkipFrameRect` = false [static]

**Warning:**
Don't load the frame rects.

Save some memory. Default MIDP 1: false
- Default MIDP 1 NOKIA: false
- Default MIDP 1 SPRINT: false
- Default MIDP 2: false
- Default MIDP 2 NOKIA: false
- Default MIDP 2 SPRINT: false
- Default DOJA: false
- Default BLACKBERRY: false Default WIPI_JAVA: false

### int `sprite_animFPS` = 25 [static]

**Warning:**
Defines the speed for sprite animations.

Set to 1000 if you use frame based timing. If this value is < 1000 then (dt) in GLLibPlayer.Update(dt) must be a Delta Time in milliseconds. If this value is = 1000 then (dt) in GLLibPlayer.Update(dt) must be a Delta Frame count. Default MIDP 1: 25
- Default MIDP 1 NOKIA: 25
- Default MIDP 1 SPRINT: 25
- Default MIDP 2: 25
- Default MIDP 2 NOKIA: 25
- Default MIDP 2 SPRINT: 25
- Default DOJA: 25
- Default BLACKBERRY: 25 Default WIPI_JAVA: 25

### boolean `sprite_debugCollision` = false [static]

**Warning:**
Use to debug sprite engine and management.
**Warning:**

Use to debug sprite engine and management.

```java
boolean sprite_debugErrors = false [static]
```

```java
boolean sprite_debugLoading = false [static]
```

```java
boolean sprite_debugModuleUsage = false [static]
```

Warning:
Use to debug sprite engine and management.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

---

boolean sprite_debugUsedMemory = false [static]

Warning:
Use to debug sprite engine and management.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

---

boolean sprite_drawPixelClippingBug = false [static]

Warning:
For some Nokia firmware, the drawPixels does not use the clipping, this option will simulate the clipping by adjusting the drawPixels parameters.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false // INVALID
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false // INVALID
Default DOJA : false // INVALID
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false // INVALID
### boolean sprite_drawRegionFlippedBug = false [static]

**Warning:**
If your device has a bugged drawRegion with flipped, this function will be replaced by drawImage.

- Default MIDP 1 : false // INVALID
- Default MIDP 1 NOKIA : false // INVALID
- Default MIDP 1 SPRINT : false // INVALID
- Default MIDP 2 : false
- Default MIDP 2 NOKIA : false
- Default MIDP 2 SPRINT : false
- Default DOJA : false // INVALID
- Default BLACKBERRY : false // INVALID
- Default WIPI_JAVA : false // INVALID

### boolean sprite_drawRGBTransparencyBug = false [static]

**Warning:**
Only use this flag when drawRGB can not draw transparency (Ex: Softbank's Samsung MIDP2 phones)

- Default MIDP 1 : false // INVALID
- Default MIDP 1 NOKIA : false // INVALID
- Default MIDP 1 SPRINT : false // INVALID
- Default MIDP 2 : false
- Default MIDP 2 NOKIA : false
- Default MIDP 2 SPRINT : false
- Default DOJA : false // INVALID
- Default BLACKBERRY : false // INVALID
- Default WIPI_JAVA : false // INVALID

### boolean sprite_fillRoundRectBug = false [static]

**Warning:**
If your device has a bugged fillRoundRect, internal call to this function will be replaced by fillRect.

- Default MIDP 1 : false
- Default MIDP 1 NOKIA : false
boolean sprite_fontBackslashChangePalette = false [static]

Warning:
When drawing a string, the '\' char is used to changed palette, it has to be followed by a number corresponding to the palette number.

boolean sprite_fontUseOneFramePerLetter = false [static]

Warning:
Each Letter Of The Font Is Mapped To Its Own Frame.

boolean sprite_fpsRegion = false [static]

Warning:
Don't use clipping when drawing sprites with drawRegion by manipulating the call's parameters.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sprite_ModuleMapping_useModuleImages = true [static]

Warning:
Each module will be exported as its own images.

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sprite_newTextRendering = true [static]

Warning:
Use new text rendering code Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sprite_RGBArraysUseDrawRGB = false [static]
Warning:  
When using sprite_ModuleMapping_useModuleImages and NOT using sprite_useCacheRGBArrays, transform images directly on the image's data in int[].

Then create a temporary image to draw it on screen. Default MIDP 1 : false // INVALID  
Default MIDP 1 NOKIA : false // INVALID  
Default MIDP 1 SPRINT : false // INVALID  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sprite_useAfOffShort = false [static]

Warning:  
Use if Aurora exported animation frames offsets as short[] instead of byte[].

Default MIDP 1 : false  
Default MIDP 1 NOKIA : false  
Default MIDP 1 SPRINT : false  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_useBSpriteFlags = false [static]

Warning:  
Use Bsprite Flags.

Must be true if sprite_usePrecomputedCRC. Default MIDP 1 : true  
Default MIDP 1 NOKIA : false // Warning, be careful  
Default MIDP 1 SPRINT : true  
Default MIDP 2 : false // Warning, be careful  
Default MIDP 2 NOKIA : false // Warning, be careful  
Default MIDP 2 SPRINT : false // Warning, be careful
**Warning:**

Some phone don't support the creation of Image with Odd width.

Using this option the Images will be created with an Even width, by padding the width to the nearest even size.

**Warning:**

Keep a version of the Single Image flipped in X and another flipped in Y.

**Warning:**

Keep the sprite's images to cache pool.
cache image to pool when draw it on screen. Use on device with small memory and slow createImage. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false // INVALID
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false // INVALID

boolean sprite_useCacheRGBArrays = false [static]

Warning:
Keep the sprite's images as int[].

Use only if your device's drawRGB is really fast. Can also be usefull if you want to manipulate the sprite's images, to create FX. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sprite_useCreateRGB = true [static]

Warning:
The sprite's Images are in RAW data format in the resources.

A palette of colors followed by a list of pixels. The internal format of the sprite's images will be Image[]. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : false
Default BLACKBERRY : true Default WIPI_JAVA : false // INVALID
### boolean sprite_useCreateRGBTransparencyBug = false [static]

**Warning:**

On some devices Image.createRGBImage the ProccessAlpha is bugged and must be always true, if tis your case, use this option.

Default MIDP 1 : false // INVALID  
Default MIDP 1 NOKIA : false // INVALID  
Default MIDP 1 SPRINT : false // INVALID  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

### boolean sprite_useDeactivateSystemGc = false [static]

**Warning:**

Use this option to disable calls to System.gc() everywhere in the sprite management.

Default MIDP 1 : false  
Default MIDP 1 NOKIA : false  
Default MIDP 1 SPRINT : false  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : true Default WIPI_JAVA : false

### boolean sprite_useDrawRegionClipping = false [static]

**Warning:**

Simulate drawregion by using drawImage and Intersection of setClip.

Default MIDP 1 : false  
Default MIDP 1 NOKIA : false
boolean sprite_useDrawStringSleep = false [static]

**Warning:**

On some low end device drawing large string on screen will choke the pipeline because there are too many drawImages, some phone will get a speed increase by doing tiny sleep at the end of each draw strings.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_useDynamicPng = false [static]

**Warning:**

The sprite's Images are Dechunked PNG in the resources.

The PNGs are recreated in memory at load time and pass to Image.createImage(byte[]); The CRC and some chunks are removed from the data to save space. Sprite's image internal format is : Image[].

Default MIDP 1 : true
Default MIDP 1 NOKIA : false // Warning, be carefull
Default MIDP 1 SPRINT : true
Default MIDP 2 : false // Warning, be carefull
Default MIDP 2 NOKIA : false // Warning, be carefull
Default MIDP 2 SPRINT : false // Warning, be carefull
Default DOJA : false // Warning, be carefull
Default BLACKBERRY : false // Warning, be carefull Default WIPI_JAVA : true
### boolean `sprite_useEncodeFormatI127RLE = true` [static]

**Warning:**
Support for Image containing 127 colors or less.

Using RLE compression. Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

### boolean `sprite_useEncodeFormatI16 = true` [static]

**Warning:**
Support for Image containing 16 colors or less.

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

### boolean `sprite_useEncodeFormatI2 = true` [static]

**Warning:**
Support for Image containing 2 colors or less.

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
<table>
<thead>
<tr>
<th>boolean sprite_useEncodeFormatI256 = false</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>Support for Image containing 256 colors or less.</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 : false</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA : false</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT : false</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 : false</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA : false</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT : false</td>
<td></td>
</tr>
<tr>
<td>Default DOJA : false</td>
<td></td>
</tr>
<tr>
<td>Default BLACKBERRY : false</td>
<td></td>
</tr>
<tr>
<td>Default WIPI_JAVA : false</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean sprite_useEncodeFormatI256RLE = true</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>Support for Image containing 256 colors or less.</td>
<td></td>
</tr>
<tr>
<td>Using RLE compression. Default MIDP 1 : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT : true</td>
<td></td>
</tr>
<tr>
<td>Default DOJA : true</td>
<td></td>
</tr>
<tr>
<td>Default BLACKBERRY : true</td>
<td></td>
</tr>
<tr>
<td>Default WIPI_JAVA : true</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean sprite_useEncodeFormatI4 = true</th>
<th>static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
<td></td>
</tr>
<tr>
<td>Support for Image containing 4 colors or less.</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA : true</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT : true</td>
<td></td>
</tr>
</tbody>
</table>
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sprite_useEncodeFormat64RLE = true [static]

Warning:
Support for Image containing 64 colors or less.

Using RLE compression. Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

boolean sprite_useExternImage = false [static]

Warning:
If you have special way to create, load, access the module's images.

Usefull when working with half transparency. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_useFMOffShort = false [static]

Warning:
Keep internal position of a module Inside a frame in short[] instead
of byte[].

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

boolean `sprite_useFMPalette` = false  [static]

**Warning:**
Use if Aurora exported the Palette to use for each module in a frame (fmodule).

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

boolean `sprite_useFrameCollRC` = false  [static]

**Warning:**
Your sprite's frames are using collision box.

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
boolean sprite_useFrameRects = false [static]

Warning:
If you have exported your sprite with frame rect, you have to load them.

Warning:
GenPalette generates a palette based on another palette.
You have to select a Flag : 1 << Flag. Available flags are:
PAL_ORIGINAL, PAL_INVISIBLE, PAL_RED_YELLOW,
PAL_BLUE_CYAN, PAL_GREEN, PAL_GREY or PAL_BLEND_BLACK

int sprite_useGenPalette = 1 << ASprite.PAL_GREY [static]

boolean sprite_useGifHeader = false [static]

Warning, be carefull
**boolean sprite_useHyperFM = false** [static]

**Warning:**
Use if your sprite contains some Hyperframe.

Hyperframes are a way to put a frame inside another frame. Default

| Default MIDP 1 | false |
| Default MIDP 1 NOKIA | false |
| Default MIDP 1 SPRINT | false |
| Default MIDP 2 | false |
| Default MIDP 2 NOKIA | false |
| Default MIDP 2 SPRINT | false |
| Default DOJA | false |
| Default BLACKBERRY | false |
| Default WIPI_JAVA | false |

**boolean sprite_useIndexExAframes = false** [static]

**Warning:**
If the sprite have more than 256 Frames per sprite you can extend to 1024 using this flag.

| Default MIDP 1 | false |
| Default MIDP 1 NOKIA | false |
| Default MIDP 1 SPRINT | false |
| Default MIDP 2 | false |
| Default MIDP 2 NOKIA | false |
| Default MIDP 2 SPRINT | false |
| Default DOJA | false |
| Default BLACKBERRY | false |
| Default WIPI_JAVA | false |

**boolean sprite_useIndexExFmodules = false** [static]

**Warning:**
If the sprite have more than 256 FModules per sprite you can extend to 1024 using this flag.
boolean `sprite_useLoadImageWithoutTransf` = false [static]

**Warning:**
Use this option if you will never use flip or rotations when drawing your sprites.

default MIDP 1 : false
default MIDP 1 NOKIA : false
default MIDP 1 SPRINT : false
default MIDP 2 : false
default MIDP 2 NOKIA : false
default MIDP 2 SPRINT : false
default DOJA : false
default BLACKBERRY : false Default WPII_JAVA : false

boolean `sprite_useModuleColorAsByte` = false [static]

**Warning:**
The color of the special module is stored as a byte instead of an int.

default MIDP 1 : false
default MIDP 1 NOKIA : false
default MIDP 1 SPRINT : false
default MIDP 2 : false
default MIDP 2 NOKIA : false
default MIDP 2 SPRINT : false
default DOJA : false
default BLACKBERRY : false Default WPII_JAVA : false

boolean `sprite_useModuleDataOffAsShort` = true [static]
**Warning:**

Use only if one of your sprite is larger than 64KB, the data will be stored in int[].

```
Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true
```

---

**boolean sprite_useModuleMapping = true [static]**

**Warning:**

Use module mapping to share data of identical sprite (except for the images data).

```
Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : false Default WIPI_JAVA : true
```

---

**boolean sprite_useModuleUsageFromSprite = false [static]**

**Warning:**

Use for recording drawing operations.

```
Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : true
Default BLACKBERRY : false Default WIPI_JAVA : false
```
<table>
<thead>
<tr>
<th><strong>boolean sprite_useModuleWHShort = false</strong> [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
</tr>
<tr>
<td>Use short instead of byte to keep module sizes.</td>
</tr>
<tr>
<td>For modules larger than -128-127. This option will take more memory.</td>
</tr>
<tr>
<td>Default MIDP 1 : false</td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA : false</td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT : false</td>
</tr>
<tr>
<td>Default MIDP 2 : false</td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA : false</td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT : false</td>
</tr>
<tr>
<td>Default DOJA : false</td>
</tr>
<tr>
<td>Default BLACKBERRY : false Default WIPI_JAVA : false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>boolean sprite_useModuleXY = false</strong> [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
</tr>
<tr>
<td>Use/Keep each module position in a Single Image.</td>
</tr>
<tr>
<td>Needed if you are using : sprite_useExternImage or sprite_useSingleImageForAllModules. Module's positions will be stored in byte[].</td>
</tr>
<tr>
<td>Default MIDP 1 : false</td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA : false</td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT : false</td>
</tr>
<tr>
<td>Default MIDP 2 : false</td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA : false</td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT : false</td>
</tr>
<tr>
<td>Default DOJA : false</td>
</tr>
<tr>
<td>Default BLACKBERRY : false Default WIPI_JAVA : false</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>boolean sprite_useModuleXYShort = false</strong> [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Warning:</strong></td>
</tr>
<tr>
<td>Use/Keep each module position in a Single Image.</td>
</tr>
<tr>
<td>Needed if you are using : sprite_useExternImage or sprite_useSingleImageForAllModules. Module's positions will be stored</td>
</tr>
</tbody>
</table>
### boolean sprite_useMultipleModuleTypes = false [static]

**Warning:**

Using this flag a module can be an image, a rectangle(MD_RECT), a solid Rectangle (MD_FILL_RECT), a Marker (MD_MARKER).

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

### boolean sprite_useNokia7650DrawPixelBug = false [static]

**Warning:**

For really Old Nokia firmware, some vertical flip where not working, this option will flip the data manually.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false // INVALID
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false // INVALID
Default DOJA : false // INVALID
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false // INVALID
**boolean sprite_useNokiaUI = false** [static]

**Warning:**
Do you want to use NokiaUI API.

The internal format of the sprite's images will be short[]. Default MIDP 1: false // INVALID
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false // INVALID
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false // INVALID
Default DOJA : false // INVALID
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false // INVALID

**boolean sprite_useNonInterlaced = false** [static]

**Warning:**
Use this flag to load the Sprite exported by Aurora with .fft config file.

NonInterlaced sprites has a good compression Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

**boolean sprite_useOperationMark = false** [static]

**Warning:**
Use for recording drawing operations.

And to mark flipped module for treatement, usefull in MIDP1 to load sprite with flipped module. Default MIDP 1 : true
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : true
### boolean sprite_useOperationRecord = false [static]

**Warning:**
Use for recording drawing operations.

<table>
<thead>
<tr>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

### boolean sprite_useOperationRect = false [static]

**Warning:**
Use for recording drawing operations.

<table>
<thead>
<tr>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>

### boolean sprite_usePixelFormat0565 = true [static]

**Warning:**
0565 Palette color encoding.

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
<th>Default DOJA</th>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
</tbody>
</table>
boolean sprite_usePixelFormat1555 = true [static]

Warning:
  1555 Palette color encoding.

boolean sprite_usePixelFormat4444 = true [static]

Warning:
  4444 Palette color encoding.

boolean sprite_usePixelFormat8888 = true [static]

Warning:
  8888 Palette color encoding.
Hardware settings for CRC

<table>
<thead>
<tr>
<th>Platform</th>
<th>Setting</th>
<th>Value</th>
<th>Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default MIDP 1</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 NOKIA</td>
<td>false</td>
<td>Warning, be careful</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 1 SPRINT</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA</td>
<td>false</td>
<td>Warning, be careful</td>
<td></td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default DOJA</td>
<td>true</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default BLACKBERRY</td>
<td>true</td>
<td>Default WIPI_JAVA: true</td>
<td></td>
</tr>
</tbody>
</table>

```java
boolean sprite_usePrecomputedCRC = false [static]
```

**Warning:**
Are the CRC exported with the image's data? For RGB or Nokia's phone, don't use this.

```java
Default MIDP 1 : true
Default MIDP 1 NOKIA : false // Warning, be careful
Default MIDP 1 SPRINT : true
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false // Warning, be careful
Default WIPI_JAVA : true
```

```java
boolean sprite_usePrecomputedFrameRect = false [static]
```

**Warning:**
Frame's Position and Size are exported in the sprite data at compile time.

Its faster but takes more memory. You need to use the BS_FRAME_RECTS flag in the .sprcmd file to export your sprites.

```java
Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false
```
boolean sprite_useResize = false [static]

Warning:
Support sprite/module resize.

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false // INVALID

boolean sprite_useSingleArrayForFMAF = false [static]

Warning:
Internal use of a single array for the module's informations and a single array for the frame's informations.

Will save some heap and code, but can slow down the game. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean sprite_useSingleDirectGraphics = false [static]

Warning:
On some Nokia devices getting the DirectGraphics can be slow or create memory leaks.

Whit this option we use only one global DirectGraphics. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false
Warning:
If your device can't load a lot (count, not size) of images, this option will use only one big image for all the modules.

Warning:
In PaintModule, check to see if the module is in part in the current clipping selection.

Use on device with slow `drawImage`. Default MIDP 1 : false

boolean *sprite_useSingleImageForAllModules* = false [static]

boolean *sprite_useSkipFastVisibilityTest* = false [static]

boolean *sprite_useTransfFlip* = false [static]
**Warning:**

When drawing a flipped Frame/Module, apply a translation to keep Frame/Module Cohesion.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : true

---

`boolean sprite_useTransfRot = false [static]`

**Warning:**

When drawing a Rotated Frame/Module, apply a translation to keep Frame/Module Cohesion.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : true

---

`boolean sprite_useTruncatedRGBBuffer = false [static]`

**Warning:**

Modify the `sprite_RGBArraysUseDrawRGB` option by drawing the image without going through a temporary Image.

This option will draw images with drawRGB. Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
**Warning:**

Used to go around some firmware bug of UTF-8 byte[] to String conversions.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : true

**Warning:**

if true text are stored in String (consume more memory, but less allocation), if false, text are stored as byte array (less memory) and converted each time a String is needed

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

**Warning:**

How many layer the tileset with the largest layer count will have.

Must be >= 1. Default MIDP 1 : 4
boolean tileset_useIndexAsShort = false [static]

Warning:
Use short instead of byte to keep tileset indexes.
For indexes larger than -128-127. This option will take more memory.

Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean tileset_useTileShift = true [static]

Warning:
Set to true if you work with power of 2 tile size (recommended).

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : true Default WIPI_JAVA : true

int TMP_BUFFER_SIZE = 256 [static]

Default MIDP 1 : 256
boolean useAbsoluteValueOfKeyCode = false [static]

Warning:
Use absolute value for keycode, if keycode can be positive or negative (as to maintain compatibility on triplets phones for example).

Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIP1_JAVA : false

boolean useBugFixMultipleKeyPressed = false [static]

Warning:
Fix the bug that some keys are not released when multiple keys are pressed.

(Something common in low end Samsung and Pantech phones).
Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIP1_JAVA : false

boolean useCallSerially = false [static]
**Warning:**
May solve some slowness/key response.

This is the only option for some phones (default = false) Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false

---

boolean **useDrawLineClippingBug** = false [static]

**Warning:**
Fix for the drawLine method bug, when it draws outside the current clip if y1 > y2 Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

---

boolean **useDrawPartialRGB** = false [static]

**Warning:**
For devices that drawing with RGB outside the screen make them crash.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false // INVALID Default WIPI_JAVA : false
### boolean useFakeInterruptHandling = false [static]

**Warning:**

For devices which don't catch interrupt properly (i.e. neither hideNotity() nor pauseApp() is called when the game is interrupted), this option will possibly help

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false // INVALID
Default WIPI_JAVA : false

### boolean useFlashLightInsteadOfVibration = false [static]

**Warning:**

If set to true, flash backLight instead of using vibration when calling Vibrate(int duration) function

Default MIDP 1 : false // INVALID
Default MIDP 1 NOKIA : false // INVALID
Default MIDP 1 SPRINT : false // INVALID
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false // INVALID
Default WIPI_JAVA : false // INVALID

### boolean useFrameDT = true [static]

**Warning:**

Use DeltaTime between each frame.

eg s_game_frameDT

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
boolean useKeyAccumulation = true [static]

**Warning:**
Key accumulation option.

Set to true if key must use accumulation buffer. Default MIDP 1: true
Default MIDP 1 NOKIA: true
Default MIDP 1 SPRINT: true
Default MIDP 2: true
Default MIDP 2 NOKIA: true
Default MIDP 2 SPRINT: true
Default DOJA: true
Default BLACKBERRY: true Default WIPI_JAVA: true

boolean useSafeDrawRegion = true [static]

**Warning:**
May prevent exceptions and slowdowns during drawing.

Slightly Default MIDP 1: true // INVALID
Default MIDP 1 NOKIA: true // INVALID
Default MIDP 1 SPRINT: true // INVALID
Default MIDP 2: true
Default MIDP 2 NOKIA: true
Default MIDP 2 SPRINT: true
Default DOJA: true
Default BLACKBERRY: true Default WIPI_JAVA: true // INVALID

boolean useSafeFillRect = false [static]

**Warning:**
Workarounds the problem of fillRect() having no effect on a backbuffer outside the size of the screen.

Slightly increases memory consumption and reduces fillRect's
**boolean useServiceRepaints = true [static]**

**Warning:**
May solve some devices will freeze if calling serviceRepaints.

This is the only option for some phones (default = true)

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
<td>:true</td>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
<td>:true</td>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default DOJA</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>:false</td>
<td>:true</td>
</tr>
</tbody>
</table>

---

**boolean useSleepInsteadOfYield = true [static]**

**Warning:**
May solve slowness if the game looks like it's using too much CPU

<table>
<thead>
<tr>
<th>Default MIDP 1</th>
<th>Default MIDP 1 NOKIA</th>
<th>Default MIDP 1 SPRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
<td>:true</td>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default MIDP 2</th>
<th>Default MIDP 2 NOKIA</th>
<th>Default MIDP 2 SPRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
<td>:true</td>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default DOJA</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default BLACKBERRY</th>
<th>Default WIPI_JAVA</th>
</tr>
</thead>
<tbody>
<tr>
<td>:true</td>
<td>:true</td>
</tr>
</tbody>
</table>

---

**boolean useSoftwareDoubleBuffer = false [static]**
Warning:
Set to true to enable a software double buffer Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean useSystemGc = true [static]

Warning:
System.gc() will be called automatically (with GLLib.Gc()) if some resource be destroyed in GLLib internal, set to false to deactivate this feature, and do it yourself.

Default MIDP 1 : true
Default MIDP 1 NOKIA : true
Default MIDP 1 SPRINT : true
Default MIDP 2 : true
Default MIDP 2 NOKIA : true
Default MIDP 2 SPRINT : true
Default DOJA : true
Default BLACKBERRY : false Default WIPI_JAVA : true

boolean xplayer_CARRIER_MXTELCEL = false [static]

Warning:
Set to true if the game is intended for MX Telcel.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false
**boolean xplayer_CARRIER_USCINGULAR_BLUE = false** [static]

**Warning:**
Set to true if the game is intended for US Cingular Blue network.

Default MIDP 1 : false  
Default MIDP 1 NOKIA : false  
Default MIDP 1 SPRINT : false  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : false  
Default WIPI_JAVA : false

**boolean xplayer_CARRIER_USCINGULAR_ORANGE = false** [static]

**Warning:**
Set to true if the game is intended for US Cingular Orange network.

Setting it to true is mandatory when using MRC. Default MIDP 1 :  
Default MIDP 1 NOKIA : false  
Default MIDP 1 SPRINT : false  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false  
Default DOJA : false  
Default BLACKBERRY : false  
Default WIPI_JAVA : false

**boolean xplayer_CARRIER_USNEXTEL = false** [static]

**Warning:**
Set to true if the game is intended for US NEXTEL.

Default MIDP 1 : false  
Default MIDP 1 NOKIA : false  
Default MIDP 1 SPRINT : false  
Default MIDP 2 : false  
Default MIDP 2 NOKIA : false  
Default MIDP 2 SPRINT : false
boolean xplayer_CARRIER_USSPRINT = false [static]

Warning:
Set to true if the game is intended for US Sprint.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean xplayer_CARRIER_USVIRGIN = false [static]

Warning:
Set to true if the game is intended for US Virgin Mobile.

Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

int xplayer_CONN_TIMEOUT = 30000 [static]

Warning:
The time in milliseconds after which a server request is timed out if a response is not received.

This is used only if xplayer_ENABLE_TIMEOUT is set to true. Default MIDP 1 : 30000
<table>
<thead>
<tr>
<th>Default MIDP 1 NOKIA</th>
<th>30000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default MIDP 1 SPRINT</td>
<td>30000</td>
</tr>
<tr>
<td>Default MIDP 2</td>
<td>30000</td>
</tr>
<tr>
<td>Default MIDP 2 NOKIA</td>
<td>30000</td>
</tr>
<tr>
<td>Default MIDP 2 SPRINT</td>
<td>30000</td>
</tr>
<tr>
<td>Default DOJA</td>
<td>30000</td>
</tr>
<tr>
<td>Default BLACKBERRY</td>
<td>3000</td>
</tr>
<tr>
<td>Default WIPI_JAVA</td>
<td>3000</td>
</tr>
</tbody>
</table>

```java
boolean xplayer_ENABLE_DEBUG = true [static]
```

**Warning:**
Enables debug print statements.

Set to false before releasing a version to either testing or submission.

- Default MIDP 1 : true
- Default MIDP 1 NOKIA : true
- Default MIDP 1 SPRINT : true
- Default MIDP 2 : true
- Default MIDP 2 NOKIA : true
- Default MIDP 2 SPRINT : true
- Default DOJA : true
- Default BLACKBERRY : true Default WIPI_JAVA : true

```java
boolean xplayer_ENABLE_DUAL_TCP = false [static]
```

**Warning:**
Enables 2 different connections for sending and receiving data using **TCP**.

Some phones don't support reading and writing using the same connection - ex. : Motorola phones. Use this only in conjunction with **ENABLE_MULTIPLAYER**.

- Default MIDP 1 : false
- Default MIDP 1 NOKIA : false
- Default MIDP 1 SPRINT : false
- Default MIDP 2 : false
- Default MIDP 2 NOKIA : false
- Default MIDP 2 SPRINT : false
- Default DOJA : false
- Default BLACKBERRY : false Default WIPI_JAVA : false
boolean xplayer_ENABLE_FIND_PLAYER = false [static]

Warning:
Enables the search player feature.

A user can be searched and his status returned. If he is currently in a session, the session name and number of players are also returned. Use this only in conjunction with ENABLE_MULTIPLAYER. Default
MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean xplayer_ENABLE_M7_SUPPORT = false [static]

Warning:
Auto defined for M7 operators.

(xplayer_CARRIER_USSPRINT | 
xplayer_CARRIER_USCINGULAR_ORANGE | 
xplayer_CARRIER_USCINGULAR_BLUE | 
xplayer_CARRIER_USNEXTEL | xplayer_CARRIER_USVIRGIN)
Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean xplayer_ENABLE_MULTIPLAYER = false [static]

Warning:
Enables support for multiplayer games using TCP connections.
Set this to true only for games having multiplayer features. If the application does not have any multiplayer features, set this to false and don’t use any of the multiplayer functions and variables, the obfuscator will remove them from the package. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

```java
boolean xplayer_ENABLE_PLAYER_SPECIFIC_DATA = false [static]

Warning:
Enables the use of player specific data for the multiplayer server.
This data is sent when the player logs on the multiplayer server, and can be retrieved by any other player using REQUEST_MESSAGE_GET_PLAYER_DATA. Use this only in conjunction with ENABLE_MULTIPLAYER. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false
```

```java
boolean xplayer_ENABLE_TIMEOUT = false [static]

Warning:
Enables or disables the outside ("software") timeout mechanism.
Use this only on phones with timeout problems (some Samsung phones). On most phones the connection times out if an answer is not received within a certain time of request start. However, on some phones, the connection thread hangs indefinitely, so we must implement an outside timeout mechanism. If set to true, the timeout
```
mechanism is enabled and the server requests timeout after
xplayer_CONN_TIMEOUT miliseconds. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean xplayer_HTTP_NO_CANCEL = false [static]

Warning:
Sets the cancel operation to either interrupt or not the connection
in progress.

On some phones, if the current connection is stopped on cancel, the
game crashes. To avoid this, the cancel operation must not interrupt
the communication and the thread should be left to finish naturally.
Also, if another connection must be performed, the application should
wait for the previous one to finish. All this can be done by setting this
constant to true. It is recommended to set this variable true on the
Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false // Warning, be careful
Default MIDP 2 NOKIA : false // Warning, be careful
Default MIDP 2 SPRINT : false // Warning, be careful
Default DOJA : false // Warning, be careful
Default BLACKBERRY : false // Warning, be careful Default
WIPI_JAVA : false

int xplayer_KEEP_ALIVE_TIME = 7000 [static]

Warning:
The idle time allowed before the TCP connection to the server is
disconnected.

Use this timer to send keep alive packets when no other transmission
being performed for a long time and the connection needs to be open.
Use this only in conjunction with ENABLE_MULTIPLAYER. Default
MIDP 1 : 7000
Default MIDP 1 NOKIA : 7000
Default MIDP 1 SPRINT : 7000
Default MIDP 2 : 7000
Default MIDP 2 NOKIA : 7000
Default MIDP 2 SPRINT : 7000
Default DOJA : 7000
Default BLACKBERRY : 7000 Default WIPI_JAVA : 7000

boolean xplayer_USE_BUG_FIX_MESSAGE_SIZE = false [static]

Warning:
Sets this if you using Sanyo phones of phones that have simmilar issue.

If we have buffer for receiveing messages with size 16 and we receive
5 bytes than some phones override the first 5 bytes and return a wrong
number of received bytes (usualy 16). This fix will set the whole buffer
to 0 and after receiving data will check how much of the bytes are
overwriten. This fix do not allow bynary data to be transfered. Set to
ture if the game is intended for Sanyo phones. Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false Default WIPI_JAVA : false

boolean xplayer_USE_HTTP_POST = false [static]

Warning:
Sets wether to use HTTP POST requests instead of GET requests
or not.

Some phones have problems if the HTTP GET request is too big.
Normally they should support more than 500 characters per request,
but some Nokia phones do not work with more than 120. On those phones, it is necessary to use **HTTP POST** requests instead of **GET** ones. Set this to true only on those phones (Nokia 6225 and related phones, some Samsung - ex.D500). Default MIDP 1 : false
Default MIDP 1 NOKIA : false
Default MIDP 1 SPRINT : false
Default MIDP 2 : false
Default MIDP 2 NOKIA : false
Default MIDP 2 SPRINT : false
Default DOJA : false
Default BLACKBERRY : false

**Warning:**

The **XPlayer** Version.

It is used to make sure the application connects to the correct server.

Default MIDP 1 : 1
Default MIDP 1 NOKIA : 1
Default MIDP 1 SPRINT : 1
Default MIDP 2 : 1
Default MIDP 2 NOKIA : 1
Default MIDP 2 SPRINT : 1
Default DOJA : 2
Default BLACKBERRY : 1
Default WIPI_JAVA : 1
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabetical List</td>
<td>Class List</td>
<td>Class Hierarchy</td>
<td>Class Members</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GLLibMidiSpectrumAnalyzer Class Reference

Author:
Nikolay Aleksiev

More...

List of all members.
## Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GLLibMidiSpectrumAnalyzer</code></td>
<td>(int bars, int maxValue, int velocityShift, int millisShift, int releaseDamping) Constructs new <code>GLLibMidiSpectrumAnalyzer</code>.</td>
</tr>
<tr>
<td><code>int MidiSpectrumAnalyzer_Parse (byte[] data)</code></td>
<td>MidiSpectrumAnalyzer_Parse the midi file.</td>
</tr>
<tr>
<td><code>void MidiSpectrumAnalyzer_Reset ()</code></td>
<td>MidiSpectrumAnalyzer_Reset the timers to 0.</td>
</tr>
<tr>
<td><code>int MidiSpectrumAnalyzer_Update (long timeMillis)</code></td>
<td>MidiSpectrumAnalyzer_Update the timer to some position in the midi song.</td>
</tr>
</tbody>
</table>
## Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int[]</td>
<td><strong>m_barLevels</strong></td>
<td>Array with values of the frequency groups.</td>
</tr>
<tr>
<td>int</td>
<td><strong>m_bars</strong></td>
<td>Number of frequency groups.</td>
</tr>
<tr>
<td>int</td>
<td><strong>m_channelFilter</strong></td>
<td>= 0x0F Filter for the channels that the analyzer will monitoring. Good practice is to disable the drums (channel 10).</td>
</tr>
<tr>
<td>long</td>
<td><strong>m_currentTime</strong></td>
<td>= 0 Current time of the analyzer. If function Snd_MediaTimeGet is not available then this parrameter can be used for the MidiSpectrumAnalyzer_Update plus delta time (analyzer.MidiSpectrumAnalyzer_Update(analyzer.m_currentTime + s_game_frameDT)).</td>
</tr>
</tbody>
</table>
### Static Public Attributes

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_INVALID_FILE_LENGTH</td>
<td>3</td>
</tr>
<tr>
<td>ERROR_INVALID_HEADER_CHUNK</td>
<td>1</td>
</tr>
<tr>
<td>ERROR_INVALID_META_EVENT</td>
<td>4</td>
</tr>
<tr>
<td>ERROR_INVALID_TRACK Chunk</td>
<td>2</td>
</tr>
<tr>
<td>NO_ERRORS</td>
<td>0</td>
</tr>
</tbody>
</table>
Detailed Description

Author: Nikolay Aleksiev
Constructs new **GLLibMidiSpectrumAnalyzer**.

**Parameters:**

- **bars**: Number of groups that the frequencies will be analyzed in. Should be value between 1 and 128.
- **maxValue**: Maximum value that is allowed for the frequency group. If the value become higher it will be clamped to the maxValue.
- **velocityShift**: Left shift that will be used for shifting the velocity of the notes. Can be used if higher values are needed.
- **millisShift**: Number of the milliseconds that passed from the last MidiSpectrumAnalyzer_Update is left shifted using this parameter and then subtract by the frequency level. If you use 64 groups you may want to left shift with one to have the same level as 128 groups.
- **releaseDamping**: Value that will be damped from the frequency level if the note is released. Looks good with 0 but can be used in some combination with millisShift.
**Member Function Documentation**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| `int MidiSpectrumAnalyzer_Parse ( byte[] data )` | MidiSpectrumAnalyzer_Parse the midi file.  
**Parameters:**  
`data` Raw data from the midi file.  
**Returns:**  
0 if no errors occurred else number of the error. |
| `void MidiSpectrumAnalyzer_Reset ( )` | MidiSpectrumAnalyzer_Reset the timers to 0. |
| `int MidiSpectrumAnalyzer_Update ( long timeMillis )` | MidiSpectrumAnalyzer_Update the timer to some position in the midi song.  
**Parameters:**  
`timeMillis` Current media time. Should be `GLLibPlayer.Snd_MediaTimeGet(…)` if `GLLib` is used.  
**Returns:**  
0 if no errors occurred else number of the error. |
**Member Data Documentation**

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final int ERROR_INVALID_FILE_LENGTH = 3</code> [static]</td>
<td></td>
</tr>
<tr>
<td><code>final int ERROR_INVALID_HEADER_CHUNK = 1</code> [static]</td>
<td></td>
</tr>
<tr>
<td><code>final int ERROR_INVALID_META_EVENT = 4</code> [static]</td>
<td></td>
</tr>
<tr>
<td><code>final int ERROR_INVALID_TRACK_CHUNK = 2</code> [static]</td>
<td></td>
</tr>
</tbody>
</table>

**m_barLevels**

Array with values of the frequency groups.

**m_bars**

Number of frequency groups.

**m_channelFilter**

Filter for the channels that the analyzer will monitoring. Good practice is to disable the drums (channel 10).

**m_currentTime**

Current time of the analyzer. If function `Snd_MediaTimeGet` is not available then this parameter can be used for the `MidiSpectrumAnalyzer_Update` plus delta time (analyzer.MidiSpectrumAnalyzer_Update(analyzer.m_currentTime +...
final int NO_ERRORS = 0 [static]
GLLibPathFinding Class Reference
[GLLibPathFinding]

A* (A star) Pathfinding class. More...

List of all members.
## Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GLLibPathFinding ()</strong></td>
<td>Empty constructor.</td>
</tr>
<tr>
<td><strong>PathFinding_Exec</strong> (int x, int y, int start_dir, int end_x, int end_y)</td>
<td>PathFinding_Exec is the heart of the engine, the search is done in this function.</td>
</tr>
<tr>
<td><strong>PathFinding_Free</strong> (boolean bKeepLastPath)</td>
<td>PathFinding_Free will free all of the internal buffers and arrays.</td>
</tr>
<tr>
<td><strong>PathFinding_GetPathLength ()</strong></td>
<td>PathFinding_GetPathLength will give you the path length that was found by the previous call to PathFinding_Exec.</td>
</tr>
<tr>
<td><strong>PathFinding_GetPathPosition (int nIndex)</strong></td>
<td>PathFinding_GetPathPosition will give you the position of the path node in array index i.e.</td>
</tr>
<tr>
<td><strong>PathFinding_GetPathPositionX (int nIndex)</strong></td>
<td>PathFinding_GetPathPositionX will give you the x position of the path node you are querying.</td>
</tr>
<tr>
<td><strong>PathFinding_GetPathPositionY (int nIndex)</strong></td>
<td>PathFinding_GetPathPositionY will give you the y position of the path node you are querying.</td>
</tr>
<tr>
<td><strong>PathFinding_Init</strong> (int nMapWidth, int nMapHeight, byte[] pPhysicalMap, int nCostMove, int nCostMoveDiag, int nCostChangeDir, int nDirCount, int nCollisionMask)</td>
<td>PathFinding_Init must be used once to initialize the PathFinding engine, or if you want to change the parameters of the search.</td>
</tr>
<tr>
<td><strong>PathFinding_Init</strong> (int nMapWidth, int nMapHeight, byte[] pPhysicalMap, int nCostMove, int nCostMoveDiag, int nCostChangeDir, int nDirCount)</td>
<td>PathFinding_Init must be used once to initialize the PathFinding engine, or if you want to change the parameters of the search.</td>
</tr>
</tbody>
</table>
# Static Public Attributes

<table>
<thead>
<tr>
<th>static final int</th>
<th>kDirDown = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td>kDirDownLeft = 6</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirDownRight = 7</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirLeft = 2</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirRight = 3</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirUp = 0</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirUpLeft = 4</td>
</tr>
<tr>
<td>static final int</td>
<td>kDirUpRight = 5</td>
</tr>
</tbody>
</table>
Detailed Description

A* (A star) Pathfinding class.

A* incrementally searches all routes leading from the starting point until it finds the shortest path to a goal. Like all informed search algorithms, it searches first the routes that appear to be most likely to lead towards the goal. What sets A* apart from a greedy best-first search is that it also takes the distance already traveled into account (the g(x) part of the heuristic is the cost from the start, and not simply the local cost from the previously expanded node).

If you want to have more information about the algorithm used in this class, see [http://www.policyalmanac.org/games/aStarTutorial.htm](http://www.policyalmanac.org/games/aStarTutorial.htm) or [http://en.wikipedia.org/wiki/A%2A](http://en.wikipedia.org/wiki/A%2A)
Constructor & Destructor Documentation

**GLLibPathFinding ( )**

Empty constructor.

**Note:**
Call PathFinding_Init to initialize the class.
Member Function Documentation

```c
void PathFinding_Exec ( int start_x,
                        int start_y,
                        int start_dir,
                        int end_x,
                        int end_y )
```

PathFinding_Exec is the heart of the engine, the search is done in this function.

The search will be completed with one call.

**Parameters:**
- `start_x` Starting tile's x coordinate.
- `start_y` Starting tile's y coordinate.
- `start_dir` Starting direction of the entity that will be moving.
- `end_x` Ending tile's x coordinate.
- `end_y` Ending tile's y coordinate.

```c
void PathFinding_Free ( boolean bKeepLastPath )
```

PathFinding_Free will free all of the internal buffers and arrays.

**Parameters:**
- `bKeepLastPath` If true, will keep the last path found array. If false, it will be freed also, and calls to query path results will fail.

```c
int PathFinding_GetPathLength ( )
```

PathFinding_GetPathLength will give you the path length that was found by the previous call to PathFinding_Exec.

**Returns:**
- The current path length. 0 if no path was found.
### Note:
You need to call `PathFinding_Exec` once before calling this function.

```c
int PathFinding_GetPathPosition ( int nIndex )
```

`PathFinding_GetPathPosition` will give you the position of the path node in array index i.e. 
\((y*\text{width}) + x\) so that you can access your array without computing the array pos.

**Parameters:**
- `nIndex` The path node you want to query.

**Returns:**
The path node position in your collision map.

**Note:**
You need to call `PathFinding_Exec` once before calling this function.

The result will be backward, going from the end to the beginning, so `PathFinding_GetPathPosition(0)` start from the end.

```c
int PathFinding_GetPathPositionX ( int nIndex )
```

`PathFinding_GetPathPositionX` will give you the x position of the path node you are querying.

**Parameters:**
- `nIndex` The path node you want to query.

**Returns:**
The X position your collision map.

**Note:**
You need to call PathFinding_Exec once before calling this fonction.

```c
int PathFinding_GetPathPositionY ( int nIndex )
```

PathFinding_GetPathPositionY will give you the y position of the path node you are querying.

**Parameters:**

- `nIndex` - The path node you want to query.

**Returns:**

The Y position your collision map.

**Note:**

You need to call PathFinding_Exec once before calling this fonction.

```c
void PathFinding_Init ( int nMapWidth,
                        int nMapHeight,
                        byte[] pPhysicalMap,
                        int nCostMove,
                        int nCostMoveDiag,
                        int nCostChangeDir,
                        int nDirCount,
                        int nCollisionMask
)
```

PathFinding_Init must be used once to initialize the PathFinding engine, or if you want to change the parameters of the search.

**Parameters:**

- `nMapWidth` - The width of your collision map grid.
- `nMapHeight` - The height of your collision map grid.
- `pPhysicalMap` - The collision map, single dimension byte array. The format is the same as a pixel array, a cell is found by cell=((y*width)+x). For the moment, the internal format of this array is 0=freecell anything else is blocked.
- `nCostMove` - is the cost of moving vertically or horizontally (but not both). Usually
**nCostMoveDiag**

is the cost of moving vertically and horizontally at the same time. Usually you should use 14. Its not used if nDirCount is 4.

**nCostChangeDir**

is the cost of changing direction. Use 0 if there is no cost involve in changing direction. Usually you should use 10.

**nDirCount**

is used to specify if the algo should look at diagonals possibilities (8) or only the minimal ones (4).

**nCollisionMask**

is a mask applied to your Physical Map to find collision. Usefull if you want to store more info in you map. By default the mask should be 0xFFFFFFFF.

```c
void PathFinding_Init ( int nMapWidth,
int nMapHeight,
byte[] pPhysicalMap,
int nCostMove,
int nCostMoveDiag,
int nCostChangeDir,
int nDirCount )
```

PathFinding_Init must be used once to initialize the PathFinding engine, or if you want to change the parameters of the search.

**Parameters:**

- **nMapWidth**
  - The width of your collision map grid.

- **nMapHeight**
  - The height of your collision map grid.

- **pPhysicalMap**
  - The collision map, single dimension byte array. The format is the same as a pixel array, a cell is found by cell=((y*width)+x). For the moment, the internal format of this array is 0=freecell anything else is blocked.

- **nCostMove**
  - is the cost of moving vertically or horizontally (but not both). Usually you should use 10.

- **nCostMoveDiag**
  - is the cost of moving vertically and horizontally at the same time. Usually you should use 14. Its not used if nDirCount is 4.

- **nCostChangeDir**
  - is the cost of changing direction. Use 0 if there is no cost involve in changing direction. Usually you should use 10.

- **nDirCount**
  - is used to specify if the algo should look at diagonals possibilities (8) or only the minimal ones (4).
Member Data Documentation

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>final int kDirDown</td>
<td>1</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirDownLeft</td>
<td>6</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirDownRight</td>
<td>7</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirLeft</td>
<td>2</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirRight</td>
<td>3</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirUp</td>
<td>0</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirUpLeft</td>
<td>4</td>
<td>[static]</td>
</tr>
<tr>
<td>final int kDirUpRight</td>
<td>5</td>
<td>[static]</td>
</tr>
</tbody>
</table>
GLLibPlayer Class Reference

[GLLibPlayer]

Generic GLLib Player. More...

Inherits Runnable.

List of all members.
## Public Member Functions

<table>
<thead>
<tr>
<th>void run ()</th>
</tr>
</thead>
</table>

*Implementation of the *Runnable* interface.*
## Public Attributes

<table>
<thead>
<tr>
<th>int</th>
<th>curTime</th>
</tr>
</thead>
</table>

*Sprite current time. Used for frame that stay on screen longer than a frame.*
Static Public Attributes

<table>
<thead>
<tr>
<th>static final int</th>
<th>WRAP_CLAMP = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>wrappng parameter, specify that the tileset is to be repeated only once</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static final int</th>
<th>WRAP_REPEAT = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>wrappng parameter, specify that the tileset is to be repeated indefinitely</td>
<td></td>
</tr>
</tbody>
</table>
### Static Protected Member Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static boolean Snd_IsPlaying(int channel)</code> throws Exception</td>
<td>Return true if a sound is currently playing on channel.</td>
</tr>
</tbody>
</table>
## Package Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final int GetAnim()</code></td>
<td>Get animation</td>
</tr>
<tr>
<td><code>final int GetDuration()</code></td>
<td>Get the duration of the current frame of the current animation.</td>
</tr>
<tr>
<td><code>final int GetFrame()</code></td>
<td>Get current frame nb</td>
</tr>
<tr>
<td><code>int GetNbanim()</code></td>
<td>Get the animation count of the current sprite.</td>
</tr>
<tr>
<td><code>int GetNbFrame()</code></td>
<td>Get the frame count of the current animation.</td>
</tr>
<tr>
<td><code>final ASprite GetSprite()</code></td>
<td>Get the current sprite.</td>
</tr>
<tr>
<td><code>final int GetTransform()</code></td>
<td>Get current sprite transformation</td>
</tr>
<tr>
<td><code>GLLibPlayer (ASprite sprite, int x, int y)</code></td>
<td>Basic constructor.</td>
</tr>
<tr>
<td><code>GLLibPlayer ()</code></td>
<td>Empty constructor.</td>
</tr>
<tr>
<td><code>boolean IsAnimOver ()</code></td>
<td>Query the state of the current animation to see if its over.</td>
</tr>
<tr>
<td><code>void Render ()</code></td>
<td>Render the current animation in GLLib current graphic context.</td>
</tr>
<tr>
<td><code>void Reset ()</code></td>
<td>Reset the current player.</td>
</tr>
<tr>
<td><code>void SetAnim (int anim, int nbLoop)</code></td>
<td>Set a new animation number to play.</td>
</tr>
<tr>
<td><code>final void SetAnim (int anim)</code></td>
<td>Set a new animation number to play, animation will loop forever.</td>
</tr>
<tr>
<td><code>int SetFrame (int frame)</code></td>
<td>Set a new frame position in the current animation.</td>
</tr>
<tr>
<td><code>final void SetPos (int x, int y)</code></td>
<td>Set new position of sprite.</td>
</tr>
<tr>
<td><code>void SetSprite (ASprite sprite)</code></td>
<td>Set a new ASprite reference in the player.</td>
</tr>
<tr>
<td><code>final void SetTransform (int transform)</code></td>
<td>Set current sprite transformation.</td>
</tr>
</tbody>
</table>
Object: `Snd_GetChannelPlayer` (int channel)

void `Update` (int DT)

*Update current animation time from 1 time unit.*

final void `Update` ()

*Update current animation time from 1 time unit.*
### Static Package Functions

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| static long | **Snd_DurationGet** (int channel) throws Exception  
*get duration of a sound* |
| static void | **Snd_ForceExecOnThreadOnGamePause** ()  
*When the game receive an interrupt, it can be important to force execution of the sound thread.* |
| static final void | **Snd_FreeChannel** (int channel)  
*Free sound channel from the player assigned to it.* |
| static int | **Snd_GetChannelVolume** (int channel) throws Exception  
*Get the current volume of a channel.* |
| static int | **Snd_GetCurrentSoundIndex** (int nChannel)  
*Get the index of the sound loaded on the channel.* |
| static void | **Snd_Init** (int nbSoundSlot) throws Exception  
*Allocate sound player and resources.* |
| static void | **Snd_LoadSound** (byte[] soundData, int nMIME, int index, boolean bCacheThisSound) throws Exception  
*Load a sound file/resource from data package.* |
| static void | **Snd_LoadSound** (byte[] soundData, int nMIME, int index) throws Exception  
*Load a sound file/resource from data package.* |
| static void | **Snd_LoadSound** (String dataFileName, int resourceIndex, boolean bCacheThisSound) throws Exception  
*Load a sound file/resource from data package.* |
| static void | **Snd_LoadSound** (String dataFileName, int resourceIndex) throws Exception  
*Load a sound file/resource from data package.* |
| static long | **Snd_MediaTimeGet** (int channel)  
*get current media time* |
| static long | **Snd_MediaTimeSet** (int channel, long time)  
*set current media time* |
| static void | **Snd_MidiPlayNote** (int channel, int MIDIChannel, int note, int volume) throws Exception  
*play a note on a midi channel.* |
| static boolean | **Snd_MidiSetChannelVolume** (int channel, int MIDIChannel, int volume) throws Exception  
*adjust volume of a midi channel* |
| static final void | **Snd_Pause** (int channel)  
*Pause sound on a channel.* |
| static void | **Snd_Play** (int channel, int index, int loop, int volume, int priority) |
Play a sound on a channel.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Snd_PrepareSound</code> (int channel, int index, int priority)</td>
<td>Prepare a sound to be played on a channel.</td>
</tr>
<tr>
<td><code>Snd_Quit</code> () throws Exception</td>
<td>Deallocate all sound resources and players.</td>
</tr>
<tr>
<td><code>Snd_RateGet</code> (int channel)</td>
<td>Get rate of a sound</td>
</tr>
<tr>
<td><code>Snd_RateGetMax</code> (int channel) throws Exception</td>
<td>Get max rate of a sound</td>
</tr>
<tr>
<td><code>Snd_RateGetMin</code> (int channel) throws Exception</td>
<td>Get min rate of a sound</td>
</tr>
<tr>
<td><code>Snd_RateSet</code> (int channel, int rate) throws Exception</td>
<td>Set rate of a sound</td>
</tr>
<tr>
<td><code>Snd_Resume</code> (int channel)</td>
<td>Resume currently paused sound on a channel.</td>
</tr>
<tr>
<td><code>Snd_SetMasterVolume</code> (int volume) throws Exception</td>
<td>Change master volume value.</td>
</tr>
<tr>
<td><code>Snd_SetMediaDuration</code> (int index, int duration) throws Exception</td>
<td>Sets duration for the sound (Use with GLLibConfig.sound_useFakeMediaDuration == true).</td>
</tr>
<tr>
<td><code>Snd_Stop</code> (int channel)</td>
<td>Stop sound on a channel.</td>
</tr>
<tr>
<td><code>Snd_StopAllSounds</code> ()</td>
<td>Stop all sounds on all channels.</td>
</tr>
<tr>
<td><code>Snd_TempoGet</code> (int channel) throws Exception</td>
<td>Get tempo of a sound</td>
</tr>
<tr>
<td><code>Snd_TempoSet</code> (int channel, int tempo) throws Exception</td>
<td>Set tempo of a sound</td>
</tr>
<tr>
<td><code>Snd_UnLoadSound</code> (int index) throws Exception</td>
<td>Unload a sound resource from memory.</td>
</tr>
<tr>
<td><code>Snd_Update</code> ()</td>
<td>Update sound engine status all sound resource.</td>
</tr>
<tr>
<td><code>Tileset_Destroy</code> (int nLayer, boolean bFreeBufferImage)</td>
<td>Delete a layer from the player.</td>
</tr>
<tr>
<td><code>Tileset_Destroy</code> (int nLayer)</td>
<td>Delete a layer from the player.</td>
</tr>
<tr>
<td><code>Tileset_Draw</code> (Graphics g, int dx, int dy, int nLayer)</td>
<td>Draw a specific onto destination Graphics.</td>
</tr>
<tr>
<td><code>Tileset_Draw</code> (Graphics g, int nLayer)</td>
<td>Draw a specific onto destination Graphics.</td>
</tr>
<tr>
<td><code>Tileset_GetBufferGraphics</code> (int p_iLayer, int p_ilImage)</td>
<td>Get</td>
</tr>
</tbody>
</table>
### Get the graphics context of one of the images being used as a buffer for this layer.

**static final Graphics**  
`Tileset_GetBufferGraphics (int p_iLayer)`

Get the graphics context of the image being used as a buffer for this layer.

### Get the graphics context of the image being used as a buffer for this layer.

**static final Image**  
`Tileset_GetBufferImage (int p_iLayer, int p_image)`

Get one of the images that is being used as a buffer for this layer.

### Get the image that is the buffer for this layer.

**static final Image**  
`Tileset_GetBufferImage (int p_iLayer)`

Get the image that is the buffer for this layer.

### Get one of the images that is being used as a buffer for this layer.

**static final int[]**  
`Tileset_GetCamera (int nLayer)`

Get the camera position of a specific layer.

### Get the camera position of a specific layer.

**static final int**  
`Tileset_GetCameraX (int nLayer)`

Get the camera X position of a specific layer.

### Get the camera X position of a specific layer.

**static final int**  
`Tileset_GetCameraY (int nLayer)`

Get the camera Y position of a specific layer.

### Get the camera Y position of a specific layer.

**static final int**  
`Tileset_GetLayerHeight (int nLayer)`

Get the pixel height of a specific layer.

### Get the pixel height of a specific layer.

**static final int**  
`Tileset_GetLayerTileCountHeight (int nLayer)`

Get the tile count height of a specific layer.

### Get the tile count height of a specific layer.

**static final int**  
`Tileset_GetLayerTileCountWidth (int nLayer)`

Get the tile count width of a specific layer.

### Get the tile count width of a specific layer.

**static final int**  
`Tileset_GetLayerWidth (int nLayer)`

Get the pixel width of a specific layer.

### Get the pixel width of a specific layer.

**static final int**  
`Tileset_GetTile (int nLayer, int x, int y)`

Get value of a specific tile.

### Get value of a specific tile.

**static final int**  
`Tileset_GetTileFlags (int nLayer, int x, int y)`

Get the flags information of a specific tile.

### Get the flags information of a specific tile.

**static void**  
`Tileset_Init (int nDestWidth, int nDestHeight, int nTileWidth, int nTileHeight)`

Initialize the GLLibPlayerTileSet engine.

### Initialize the GLLibPlayerTileSet engine.

**static void**  
`Tileset_LoadLayer (int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, int iUseCB, int origin, int wrappingX, int wrappingY)`

Load a tileset layer into the player.

### Load a tileset layer into the player.

**static void**  
`Tileset_LoadLayer (int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, boolean bUseCB, int origin, int wrappingX, int wrappingY)`

Load a tileset layer into the player.

### Load a tileset layer into the player.

**static final void**  
`Tileset_SetCamera (int nLayer, int x, int y)`

Set the camera position of a specific layer.

### Set the camera position of a specific layer.

**static void**  
`Tileset_Update (int nLayer)`

Update a specific layer circular buffer (back buffer) but does nothing if
this layer is not using a circular buffer.
### Package Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>curFlags</td>
<td>Sprite player current display flags.</td>
</tr>
<tr>
<td>int</td>
<td>posX</td>
<td>Sprite player current X position.</td>
</tr>
<tr>
<td>int</td>
<td>posY</td>
<td>Sprite player current Y position.</td>
</tr>
<tr>
<td>ASprite</td>
<td>sprite</td>
<td>Sprite player current ASprite reference.</td>
</tr>
</tbody>
</table>
## Static Package Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| static final int | k_snd_priority_highest | 0  
**Sound priorities constant.** |
| static final int | k_snd_priority_lowest | 15  
**Sound priorities constant.** |
| static final int | k_snd_priority_normal | 7  
**Sound priorities constant.** |
| static String[] | s_data_mimeType |  
*Sound Mime Container, to know the sound type of each slots.* |
| static int | s_snd_masterVolume |  
*Sound engine master volume (0-100). The volume will be scaled to the phone range.* |
| static int | s_snd_maxNbSoundSlot |  
*Sound maximum slot count for this game. The maximum number of sounds that can be loaded into memory.* |
Detailed Description

Generic GLLib Player.

This class is used to Play ASprite animations and to play Musics/Sounds FX.

The Animation part of the player will use ASprite animation. You can have many instances of the player using the same ASprite to keep track of each animation state, progression and position.

The Sound part of the player is unique so it does not have to be instanciated. Once initiated, you can access it statically from everywhere in your code.
ATileSet Aurora TileSet class. More...

List of all members.
## Static Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final int</td>
<td><code>BOTTOM</code> = Graphics.BOTTOM</td>
</tr>
<tr>
<td>static final int</td>
<td><code>HCENTER</code> = Graphics.HCENTER</td>
</tr>
<tr>
<td>static final int</td>
<td><code>HCENTER_BOTTOM</code> = Graphics.HCENTER</td>
</tr>
<tr>
<td>static final int</td>
<td><code>HCENTER_TOP</code> = Graphics.HCENTER</td>
</tr>
<tr>
<td>static final int</td>
<td><code>HCENTER_VCENTER</code> = Graphics.HCENTER</td>
</tr>
<tr>
<td>static final int</td>
<td><code>LEFT</code> = Graphics.LEFT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>LEFT_BOTTOM</code> = Graphics.LEFT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>LEFT_TOP</code> = Graphics.LEFT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>LEFT_VCENTER</code> = Graphics.LEFT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>RIGHT</code> = Graphics.RIGHT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>RIGHT_BOTTOM</code> = Graphics.RIGHT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>RIGHT_TOP</code> = Graphics.RIGHT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>RIGHT_VCENTER</code> = Graphics.RIGHT</td>
</tr>
<tr>
<td>static final int</td>
<td><code>TOP</code> = Graphics.TOP</td>
</tr>
<tr>
<td>static final int</td>
<td><code>VCENTER</code> = Graphics.VCENTER</td>
</tr>
</tbody>
</table>
Detailed Description

ATileSet Aurora TileSet class.

Implementation for Tile Set (aTLMap) exported by AuroraGT editor. Contains methods to create and display the map using a circular buffer.
Member Data Documentation

final int BOTTOM = Graphics.BELOW [static]

final int HCENTER = Graphics.CENTER [static]

final int HCENTER_BOTTOM = Graphics.CENTER | Graphics.BELOW [static]

final int HCENTER_TOP = Graphics.CENTER | Graphics.ABOVE [static]

final int HCENTER_VCENTER = Graphics.CENTER | Graphics.MIDDLE [static]

final int LEFT = Graphics.LEFT [static]

final int LEFT_BOTTOM = Graphics.LEFT | Graphics.BELOW [static]

final int LEFT_TOP = Graphics.LEFT | Graphics.ABOVE [static]

final int LEFT_VCENTER = Graphics.LEFT | Graphics.MIDDLE [static]

final int RIGHT = Graphics.RIGHT [static]

final int RIGHT_BOTTOM = Graphics.RIGHT | Graphics.BELOW [static]
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`final int RIGHT_TOP = Graphics.RIGHT</td>
<td>Graphics.TOP [static]`</td>
</tr>
<tr>
<td>`final int RIGHT_VCENTER = Graphics.RIGHT</td>
<td>Graphics.VCENTER [static]`</td>
</tr>
<tr>
<td><code>final int TOP = Graphics.TOP [static]</code></td>
<td></td>
</tr>
<tr>
<td><code>final int VCENTER = Graphics.VCENTER [static]</code></td>
<td></td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:32 2008 for GLib by doxygen 1.5.2*
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabetical List</td>
<td>Class List</td>
<td>Class Hierarchy</td>
<td>Class Members</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HTTP Class Reference
[HTTP (used by XPlayer and License)]

Inherits Runnable.

[List of all members.]
## Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cancel ()</code></td>
<td>Cancels the ongoing transaction.</td>
</tr>
<tr>
<td><code>cleanup ()</code></td>
<td>Frees any resources used by this class.</td>
</tr>
<tr>
<td><code>isErrorOccurred ()</code></td>
<td>Tells whether an error occurred during the transaction.</td>
</tr>
<tr>
<td><code>isInProgress ()</code></td>
<td>Provides information whether this class' instance is currently doing a transfer or not.</td>
</tr>
<tr>
<td><code>run ()</code></td>
<td>The main thread handling the <strong>HTTP</strong> request.</td>
</tr>
</tbody>
</table>
| `sendByGet (String sUrl, String sQuery)` | The entry point. }
## Public Attributes

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td><strong>m_bError</strong> = false</td>
</tr>
<tr>
<td></td>
<td><em>Tells whether an error occurred during the HTTP transaction.</em></td>
</tr>
<tr>
<td>String</td>
<td><strong>m_response</strong></td>
</tr>
<tr>
<td></td>
<td><em>The received message buffer.</em></td>
</tr>
<tr>
<td>byte[]</td>
<td><strong>m_responseByteArray</strong></td>
</tr>
<tr>
<td></td>
<td><em>The received message buffer in byte array.</em></td>
</tr>
</tbody>
</table>
# Static Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td><code>CarrierDeviceId</code> = null</td>
<td>Stores the emulated value of the CarrierDeviceId when running on an emulator.</td>
</tr>
<tr>
<td>static</td>
<td><code>clientId</code> = null</td>
<td>Stores the emulated value of the ClientId when running on an emulator.</td>
</tr>
<tr>
<td>static</td>
<td><code>upsubId</code> = null</td>
<td>Stores the emulated value of the UPSUBID when running on an emulator.</td>
</tr>
<tr>
<td>static</td>
<td><code>userAgent</code> = null</td>
<td>The User-Agent to be used.</td>
</tr>
<tr>
<td>static</td>
<td><code>x_up_calling_line_id</code> = null</td>
<td>Stores the emulated value of the x_up_calling_line_id when running on an emulator.</td>
</tr>
<tr>
<td>static</td>
<td><code>x_up_subno</code> = null</td>
<td>Stores the emulated value of the x_up_subno when running on an emulator.</td>
</tr>
</tbody>
</table>
Member Function Documentation

void cancel()

Cancels the ongoing transaction.

It is safe to call at any time, but keep in mind that the request may have already reached the server, triggering any changes it implied. Those changes will not be discarded by the server and will be saved.

void cleanup()

Frees any resources used by this class.

It is OK to call at any time, but it should be called after using any data returned from the server as it also deletes m_response.

Side effect:
   It will cancel the current transaction, if any.

See also:
   m_response

boolean isErrorOccurred()

Tells wether an error occurred during the transaction.

Returns:
   true if an error has been encountered, false otherwise.

boolean isInProgress()

Provides information wether this class' instance is currently doing a transfer or not.
Returns:
true if a call is in progress, false otherwise.

```java
void run ()
```

The main thread handling the HTTP request.

It is started by the `sendByGet(String, String)` function.

See also:
`sendByGet(String, String)`

```java
void sendByGet ( String sUrl,
                String sQuery
              )
```

The entry point.

It starts a HTTP request.

Parameters:
- `sUrl` URL of the server
- `sQuery` The query to be sent to the server
Member Data Documentation

String **CarrierDeviceId** = null [static]

Stores the emulated value of the CarrierDeviceId when running on an emulator.

Each Cingular Orange Network phone comes with a factory preset unique CarrierDeviceId. This is not true for the emulator though, which does not have such a value associated with it.

Because this value is requested by the server, for testing on emulator this needs to be emulated somehow. To do this, set the **CarrierDeviceId** property in the .jad file to something like "YourName01@gameloft.com" to enable sending of a CarrierDeviceId while debugging in emulator. The property value will be read from the .jad, set accordingly in here and used throughout the running of the transaction.

DO NOT set this in the .jad on a phone release though. The phone's CarrierDeviceId will be automatically added by the carrier.

Default is null.

---

String **clientId** = null [static]

Stores the emulated value of the ClientId when running on an emulator.

Each Sprint phone comes with a factory preset unique ClientId. This is not true for the emulator though, which does not have such a value associated with it.

Because this value is requested by the server, for testing on emulator this needs to be emulated somehow. To do this, set the **ClientId** property in the .jad file to something like "SZZYourName01@sprintpcs.com" to enable sending of a client id while debugging in emulator. The property value will be read from the
.jad, set accordingly in here and used throughout the running of the transaction.

DO NOT do this on a phone release though. The phone's ClientId will be automatically used there.

Default is null.

```java
boolean m_bError = false
```

Tells whether an error occurred during the HTTP transaction.

It is set to true if an error has been encountered and false otherwise.

```java
String m_response
```

The received message buffer.

Check this after a transaction completes to get the received data.

```java
byte[] m_responseByteArray
```

The received message buffer in byte array.

Check this after a transaction completes to get the received data.

```java
String upsbid = null [static]
```

Stores the emulated value of the UPSUBID when running on an emulator.

Each Nextel Network phone comes with a factory preset unique UPSUBID. This is not true for the emulator though, which does not have such a value associated with it.
Because this value is requested by the server, for testing on emulator this needs to be emulated somehow. To do this, set the **UPSUBID** property in the .jad file to something like "YourName01@gameloft.com" to enable sending of a UPSUBID while debugging in emulator. The property value will be read from the .jad, set accordingly in here and used throughout the running of the transaction.

DO NOT do this on a phone release though. The phone's UPSUBID will be automatically used there.

Default is **null**.

---

**String userAgent = null [static]**

The User-Agent to be used.

Default is "GameloftClient/1.0"; set to whatever User-Agent you need. If **null**, no user agent will be set (device defaults will be used instead).

---

**String x_up_calling_line_id = null [static]**

Stores the emulated value of the x_up_calling_line_id when running on an emulator.

For testing Virgin Mobile games only. Set **x_up_calling_line_id** in .jad to a 10 digit phone number to enable sending of a x_up_calling_line_id while debugging in emulator.

Default is **null**.

---

**String x_up_subno = null [static]**

Stores the emulated value of the x_up_subno when running on an emulator.

Each Cingular Blue Network phone comes with a factory preset unique
x_up_subno. This is not true for the emulator though, which does not have such a value associated with it.

Because this value is requested by the server, for testing on emulator this needs to be emulated somehow. To do this, set the **x_up_subno** property in the .jad file to something like "YourName01@gameloft.com" to enable sending of a client id while debugging in emulator. The property value will be read from the .jad, set accordingly in here and used throughout the running of the transaction.

DO NOT set this in the .jad on a phone release though. The phone's x_up_subno will be automatically added by the carrier.

Default is **null**.
License Class Reference

The base class for License communication with the Gameloft server.

More...

List of all members.
# Public Member Functions

<table>
<thead>
<tr>
<th>void</th>
<th>cancel ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cancels the ongoing request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th>cleanup ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cleans up the memory allocated during the last server request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>getLastError ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the status of the last request.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th>handleValidateLicense ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cingular Specific - Handles the license validation request, after such a request has been initiated with sendValidateLicense(), this function's asynchronous pair.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th>isLicenseValid ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>(MIDlet midlet)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th>sendValidateLicense ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cingular Specific - Sends a request to retrace the users license expiration time.</td>
</tr>
</tbody>
</table>
### Static Public Attributes

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static long</td>
<td><strong>callstarttime</strong>&lt;br&gt;The time the current request started - used for timeout implementation.</td>
</tr>
<tr>
<td>static final int</td>
<td><strong>NOT_A_NUMBER</strong> = -666666&lt;br&gt;Used to define anything that should be a number but it has not been initialised or does not have a value yet.</td>
</tr>
</tbody>
</table>
# Classes

| interface | Error License error codes. More... |
Detailed Description

The base class for License communication with the Gameloft server.

Important notice : make sure the API connects to the corresponding servers!!! Ask interactivity team for the correct urls.

Control the set of features offered by the api using the Configuration interface.
The api consists of: Cingular License Validation

License

Provides services for validating a Cingular users subscription.

The main mechanism for passing requests to the Gameloft server is by using HTTP transactions. Every action that results in contacting the server has a pair of asynchronous functions, a send function and a handle.

The send function builds up the HTTP message with the request and initiates the transaction (sends that message to the server).

The handle manages the connection and any events that might occur, for example a response from the server, a timeout or an error. It sets an error code indicating the status of the current connection which can be viewed by the means of the getLastError() function provided.

This means that the usual cycle of a request is the following:
1. initiate the transaction by sending the request
2. while the result from the server is pending and a timeout has not occured, call the handle and check the error code returned with getLastError();
3. analysing the error code returned, determine a correct response (if the result is still pending, wait some more, if timedout or some other error display a meaningful message, etc.)

NOTE: Since subscriptions will not work when testing the games the
testing server will randomly return:

- Valid for 12 hours 80% of the time.
- **License** Expired 25% of the time.
- **Error** 5% of the time.

Following is a short example of a generic game **License** Validation code:

The process for license validation is different than that of the standard XPlayer implementation. This is because it has to run before the game even start and has to be as efficient as possible. The code that should be add before the game is started follows:

```java
License xplayer = new License(midlet_);
```

Check to see if the license stored in RMS is valid.
```java
if (xplayer.isLicenseValid())
{
    The license is good!
    ...// TODO: Start the game.
    System.out.println("License valid, start the game!");
}
else
{
    Check with server to see if MRC has been renewed.
    xplayer.sendValidateLicense();
}
```

We call the handle and get the error code returned by it.
```java
    while (xplayer.handleValidateLicense() != true)
    {
        Wait for the response.
        ...// TODO: Here there should be a test on a keypress event would call the cancel() method of the XPlayer class, canceling any request; this would provide the user with an option to cancel the transaction.
    }
```

Get the error.
```java
    int _errCode = xplayer.getLastErrorCode();
```

xplayer.cleanup();

Verify response from server.
```java
    switch (_errCode)
```
{  
    case XPlayer.Error.ERROR_NONE:  
        Successful response.  
        Check to see if the new license stored in RMS is valid.  
        if (xplayer.isLicenseValid())  
            The license is good!  
                ...// TODO: Start the game.  
                System.out.println("License valid, start the game!");  
        }  
        else  
            License has expired.  
                ...// TODO: Let the user know and don't start the game  
                System.out.println("License Expired.");  
        }  
        break;

    case XPlayer.Error.ERROR_CONNECTION:  
        A connection error occurred.  
        And either the timestamp in RMS has expired or the RMS doesn't exist.  
                ...// TODO: An error message should be displayed and an option to retry provided.  
                System.out.println("Connection error. Please try again later.");  
        break;

    default:  
        Some other error occurred.  
        And either the timestamp in RMS has expired or the RMS doesn't exist.  
                ...// TODO: An error message should be displayed and an option to retry provided.  
                System.out.println("Some other network error. Please try again later.");  
        break;
  
} // End of else

Author:  
    Gameloft Interactivity Team

Version:  
    1.0.0
### Constructor & Destructor Documentation

**License** *(MIDlet* `midlet`*)

Constructor for the `XPlayer` class.

It reads the values of the `XPlayerURL` and `GGI` properties from the `.jad` file of `midlet` for initialisation purposes.

**Parameters:**

- `midlet`  Reference to the MIDlet of the game which implements this API.
void cancel ()

Cancels the ongoing request.

void cleanup ()

Cleans up the memory allocated during the last server request.

Use this everytime a request completes (it finishes being pending - check with getLastError()) to free up some memory.

See also: getLastError()

int getLastError ()

Gets the status of the last request.

Call this after any handle function to find out the status of the request.

If ERROR_PENDING is returne, the call is still in progress.

Returns: One of the error codes defined in the XPlayer.Error interface depending on the last request's status.

See also: XPlayer.Error

boolean handleValidateLicense ()

Cingular Specific - Handles the license validation request, after such a request has been initiated with sendValidateLicense(), this function's
asynchronous pair.

This function is valid only for Cingular games

This license validation is only for Cingular games. This is not supported for other operators so it shouldn't be used on them.

See also:
  handleValidateLicense()

boolean isLicenseValid ( )

void sendValidateLicense ( )

Cingular Specific - Sends a request to retrieve the users license expiration time.

This function is asynchronous and valid only for Cingular games.

Use handleValidateLicense() to handle the request afterwards.

This license validation is only for Cingular games. This is not supported for other operators so it shouldn't be used on them.

See also:
  sendValidateLicense()
### Member Data Documentation

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td><code>callstarttime</code></td>
<td>The time the current request started - used for timeout implementation. Use only in conjunction with ENABLE_TIMEOUT.</td>
</tr>
<tr>
<td>final int</td>
<td><code>NOT_A_NUMBER</code></td>
<td>Used to define anything that should be a number but it has not been initialised or does not have a value yet. For example when the user inquires about his score for a level that he has never played yet or anything similar.</td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:32 2008 for GLib by doxygen 1.5.2*
License.Error Interface Reference

License error codes. More...

List of all members.
## Static Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Attribute Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final</td>
<td><code>ERROR_BAD_RESPONSE</code> = 40</td>
<td></td>
<td>The server response has incorrect format.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_CONNECTION</code> = -2</td>
<td></td>
<td>No connection to server.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_INIT</code> = -100</td>
<td></td>
<td>Initialisation value.</td>
</tr>
<tr>
<td>int</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_INVALID_GGI</code> = 27</td>
<td></td>
<td>Configuration error: Most likely an invalid GGI.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_NO_CLIENT_ID</code> = 26</td>
<td></td>
<td>Programming error on emulator: no phone id.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_NO_PHONE_NUMBER</code> = 25</td>
<td></td>
<td>There is no phone number in the register request.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_NO_UUID</code> = 1</td>
<td></td>
<td>The uuid is missing from the request.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_NONE</code> = 0</td>
<td></td>
<td>The request finished successfully.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static final</td>
<td><code>ERROR_PENDING</code> = -1</td>
<td></td>
<td>Result is pending.</td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

License error codes.

This interface contains defines for error codes. It is not necessary that the application treat all these error codes, but they are very useful for debugging.
## Member Data Documentation

<table>
<thead>
<tr>
<th>final byte</th>
<th>ERROR_BAD_RESPONSE = 40 [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The server response has incorrect format.</td>
</tr>
<tr>
<td></td>
<td>Make sure the request is correct. If it is a leaderboard request, make sure the number of supplemental data needed match the requested leaderboard</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>final byte</th>
<th>ERROR_CONNECTION = -2 [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No connection to server.</td>
</tr>
<tr>
<td></td>
<td>Either there is no network available or the server cannot be contacted for some reason.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>final int</th>
<th>ERROR_INIT = -100 [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initialisation value.</td>
</tr>
<tr>
<td></td>
<td>Getting this error means that actually no requests have been yet made since the MIDlet started so there is no last error.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>final byte</th>
<th>ERROR_INVALID_GGI = 27 [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configuration error: Most likely an invalid GGI.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>final byte</th>
<th>ERROR_NO_CLIENT_ID = 26 [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Programming error on emulator: no phone id.</td>
</tr>
</tbody>
</table>
There is no phone number in the register request.

The uuid is missing from the request.

The request finished successfully.

Result is pending.

Transaction with server is ongoing.
MessageType Interface Reference

List of all members.
## Static Public Attributes

<table>
<thead>
<tr>
<th>Static final byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final byte</td>
<td><strong>BASIC_MESSAGE_TYPE_CONNECT</strong> = 's'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>BASIC_MESSAGE_TYPE_GAME</strong> = 'g'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>BASIC_MESSAGE_TYPE_UNKNOWN</strong> = 0</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>CONNECT_MESSAGE_DISCONNECT</strong> = 'x'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>CONNECT_MESSAGE_ERROR</strong> = 'e'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>CONNECT_MESSAGE_INIT_READ</strong> = 'r'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>CONNECT_MESSAGE_INIT_WRITE</strong> = 'w'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>CONNECT_MESSAGE_SUCCESS</strong> = 's'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_BLOCKED_SESSION</strong> = 'b'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_CREATE_SESSION_INVALID_NAME</strong> = 'v'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_CREATE_SESSION_NAME_USED</strong> = 'u'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_INVALID_INPUT</strong> = 'i'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</strong> = 'j'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_KICK_OUT_PLAYER_IS_MASTER</strong> = 'd'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_LIST_SESSION_INVALID_INDEX</strong> = 'l'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_LOGIN_AUTHENTICATION_FAILED</strong> = 'a'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_LOGIN_INVALID_NICKNAME</strong> = 'q'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_LOGIN_NICKNAME_USED</strong> = 'n'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_NO_PLAYER</strong> = 'k'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_NO_SESSION</strong> = 's'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_NOT_MASTER</strong> = 'm'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_SESSION_CLOSED</strong> = 'c'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</strong> = 'g'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_ERROR</strong> = 'e'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_IN_GAME</strong> = 'g'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_IN_GAME_TOALL</strong> = 'h'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_KEEP_ALIVE</strong> = 'a'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_PUSH</strong> = 'p'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>GAME_MESSAGE_REQUEST</strong> = 'r'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_HAS_NO_PSD</strong> = 'd'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_HAS_PSD</strong> = 'h'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_NOT_REGISTERED</strong> = 'n'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_OFFLINE</strong> = 'f'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_ONLINE</strong> = 'o'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_ONLINE_IN_SESSION</strong> = 's'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PLAYER_STATUS_ONLINE_PLAYING</strong> = 'p'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PUSH_MESSAGE_FINISH_GAME</strong> = 'f'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PUSH_MESSAGE_JOIN_SESSION</strong> = 'j'</td>
</tr>
<tr>
<td>static final byte</td>
<td><strong>PUSH_MESSAGE_KICK_OUT</strong> = 'k'</td>
</tr>
<tr>
<td>static final byte</td>
<td>PUSH_MESSAGE_LEAVE_SESSION = 'l'</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>static final byte</td>
<td>PUSH_MESSAGE_START_GAME = 's'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_CREATE_SESSION = 'c'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_FINISH_GAME = 'f'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_GET_PLAYER_DATA = 'y'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_GET_PLAYER_INFO = 'w'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_GET_QUICK_SESSION = 'u'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_JOIN_SESSION = 'j'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_KICK_OUT_PLAYER = 'k'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_LEAVE_SESSION = 'q'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_LIST_SESSIONS = 'l'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_LOGIN = 'i'</td>
</tr>
<tr>
<td>static final byte</td>
<td>REQUEST_MESSAGE_PROXY = 'g'</td>
</tr>
<tr>
<td>static final byte</td>
<td>RESPONSE_MESSAGE_ERROR = 'e'</td>
</tr>
<tr>
<td>static final byte</td>
<td>RESPONSE_MESSAGE_SUCCESS = 's'</td>
</tr>
</tbody>
</table>
Member Data Documentation

<table>
<thead>
<tr>
<th>Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>final byte BASIC_MESSAGE_TYPE_CONNECT = 's' [static]</td>
</tr>
<tr>
<td>final byte BASIC_MESSAGE_TYPE_GAME = 'g' [static]</td>
</tr>
<tr>
<td>final byte BASIC_MESSAGE_TYPE_UNKNOWN = 0 [static]</td>
</tr>
<tr>
<td>final byte CONNECT_MESSAGE_DISCONNECT = 'x' [static]</td>
</tr>
<tr>
<td>final byte CONNECT_MESSAGE_ERROR = 'e' [static]</td>
</tr>
<tr>
<td>final byte CONNECT_MESSAGE_INIT_READ = 'r' [static]</td>
</tr>
<tr>
<td>final byte CONNECT_MESSAGE_INIT_WRITE = 'w' [static]</td>
</tr>
<tr>
<td>final byte CONNECT_MESSAGE_SUCCESS = 's' [static]</td>
</tr>
<tr>
<td>final byte ERROR_BLOCKED_SESSION = 'b' [static]</td>
</tr>
<tr>
<td>final byte ERROR_CREATE_SESSION_INVALID_NAME = 'v' [static]</td>
</tr>
<tr>
<td>final byte ERROR_CREATE_SESSION_NAME_USED = 'u' [static]</td>
</tr>
<tr>
<td>Error Code</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>ERROR_INVALID_INPUT</td>
</tr>
<tr>
<td>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</td>
</tr>
<tr>
<td>ERROR_KICK_OUT_PLAYER_IS_MASTER</td>
</tr>
<tr>
<td>ERROR_LIST_SESSION_INVALID_INDEX</td>
</tr>
<tr>
<td>ERROR_LOGIN_AUTHENTICATION_FAILED</td>
</tr>
<tr>
<td>ERROR_LOGIN_INVALID_NICKNAME</td>
</tr>
<tr>
<td>ERROR_LOGIN_NICKNAME_USED</td>
</tr>
<tr>
<td>ERROR_NO_PLAYER</td>
</tr>
<tr>
<td>ERROR_NO_SESSION</td>
</tr>
<tr>
<td>ERROR_NOT_MASTER</td>
</tr>
<tr>
<td>ERROR_SESSION_CLOSED</td>
</tr>
<tr>
<td>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</td>
</tr>
<tr>
<td>Variable</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_ERROR</code></td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_IN_GAME</code></td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_IN_GAME_TOALL</code></td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_KEEP_ALIVE</code></td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_PUSH</code></td>
</tr>
<tr>
<td>final byte <code>GAME_MESSAGE_REQUEST</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_HAS_NO_PSD</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_HAS_PSD</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_NOT_REGISTERED</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_OFFLINE</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_ONLINE</code></td>
</tr>
<tr>
<td>final byte <code>PLAYER_STATUS_ONLINE_IN_SESSION</code></td>
</tr>
<tr>
<td>Message Type</td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td>PLAYER_STATUS_ONLINE_PLAYING</td>
</tr>
<tr>
<td>PUSH_MESSAGE_FINISH_GAME</td>
</tr>
<tr>
<td>PUSH_MESSAGE_JOIN_SESSION</td>
</tr>
<tr>
<td>PUSH_MESSAGE_KICK_OUT</td>
</tr>
<tr>
<td>PUSH_MESSAGE_LEAVE_SESSION</td>
</tr>
<tr>
<td>PUSH_MESSAGE_START_GAME</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_CREATE_SESSION</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_FINISH_GAME</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_DATA</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_INFO</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_QUICK_SESSION</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_JOIN_SESSION</td>
</tr>
<tr>
<td>Message Type</td>
</tr>
<tr>
<td>------------------------------------</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_KICK_OUT_PLAYER</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LEAVE_SESSION</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LIST_SESSIONS</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LOGIN</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_PROXY</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_START_GAME</td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_ERROR</td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_SUCCESS</td>
</tr>
</tbody>
</table>
Runnable Class Reference

Inherited by GLLib, GLLibPlayer, and HTTP.
TCP Class Reference

[TCP (used by XPlayer)]

Inherits Thread.

List of all members.
## Public Member Functions

<table>
<thead>
<tr>
<th>void</th>
<th><strong>connect</strong> ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Establish the connection.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th><strong>disconnect</strong> ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Close the connection.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte[]</th>
<th><strong>recvPacket</strong> ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Gets the first available packet from receiving queue, or null.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th><strong>run</strong> ()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Main running function for the communication thread.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th><strong>sendEstablishConnectionPackageOnReceive</strong> (byte[] data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Sends the establish message on the receive connection.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th><strong>sendPacket</strong> (byte[] data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Puts the data as packet to the sending queue.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TCP</strong></th>
<th>(String url)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Constructor for the TCP class.</em></td>
</tr>
</tbody>
</table>
**Public Attributes**

<table>
<thead>
<tr>
<th>Type</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>m_bError</td>
<td>Tells whether an error occurred during the TCP/IP transaction.</td>
</tr>
<tr>
<td>String</td>
<td>m_bErrorString</td>
<td>Error string, used only when debugging is enabled.</td>
</tr>
<tr>
<td>boolean</td>
<td>m_connected</td>
<td>true if the connection is established with the multiplayer server, false otherwise.</td>
</tr>
</tbody>
</table>
**Protected Attributes**

<table>
<thead>
<tr>
<th>Class</th>
<th>Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamConnection</td>
<td>connection = null</td>
</tr>
<tr>
<td>Vector</td>
<td>incomingQueue</td>
</tr>
<tr>
<td>InputStream</td>
<td>istream = null</td>
</tr>
<tr>
<td>OutputStream</td>
<td>ostream = null</td>
</tr>
<tr>
<td>byte[]</td>
<td>outgoing_data_on_receive_connection</td>
</tr>
<tr>
<td>Vector</td>
<td>outgoingQueue</td>
</tr>
<tr>
<td>StreamConnection</td>
<td>receive_connection = null</td>
</tr>
<tr>
<td>InputStream</td>
<td>receive_istream = null</td>
</tr>
<tr>
<td>OutputStream</td>
<td>receive_ostream = null</td>
</tr>
<tr>
<td>String</td>
<td>url</td>
</tr>
</tbody>
</table>
Constructor & Destructor Documentation

**TCP (String *url*)**

Constructor for the **TCP** class.

**Parameters:**

*url* The multiplayer server url.
## Member Function Documentation

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void connect()</code></td>
<td>Establish the connection.</td>
</tr>
<tr>
<td><code>void disconnect()</code></td>
<td>Close the connection.</td>
</tr>
<tr>
<td><code>byte[] recvPacket()</code></td>
<td>Gets the first available packet from receiving queue, or null. <strong>Returns:</strong> byte[]</td>
</tr>
<tr>
<td><code>void run()</code></td>
<td>Main running function for the communication thread.</td>
</tr>
<tr>
<td><code>void sendEstablishConnectionPackageOnReceive(byte[] data)</code></td>
<td>Sends the establish message on the receive connection. In order to establish a connection, two packages must be sent one on the receive connection and the other on the send connection. The server will match them and create a new connection. <strong>Parameters:</strong> data byte[]</td>
</tr>
</tbody>
</table>
void sendPacket ( byte[] data )

Puts the data as packet to the sending queue.

Parameters:
  data  byte[]
**Member Data Documentation**

- **StreamConnection connection** = null [protected]
- **Vector incomingQueue** [protected]
- **InputStream istream** = null [protected]
- **boolean m_bError**
  
  Tells whether an error occurred during the TCP/IP transaction.
  
  It is set to `true` if there was an error in the communication with the multiplayer server, `false` otherwise.
- **String m_bErrorString**
  
  Error string, used only when debugging is enabled.
- **boolean m_connected**
  
  `true` if the connection is established with the multiplayer server, `false` otherwise.
- **OutputStream ostream** = null [protected]
- **byte[] outgoing_data_on_receive_connection** [protected]
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Vector outgoingQueue</code></td>
<td>[protected]</td>
<td></td>
</tr>
<tr>
<td><code>StreamConnection receive_connection</code></td>
<td>= null [protected]</td>
<td></td>
</tr>
<tr>
<td><code>InputStream receive_istream</code></td>
<td>= null [protected]</td>
<td></td>
</tr>
<tr>
<td><code>OutputStream receive_ostream</code></td>
<td>= null [protected]</td>
<td></td>
</tr>
<tr>
<td><code>String url</code></td>
<td>[protected]</td>
<td></td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:33 2008 for GLib by doxygen 1.5.2*
XPlayer Class Reference

[XPlayer core]

List of all members.
# Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| **void** | **addByte** (byte data) throws Exception  
*Multiplayer Specific - Add a byte to the current packet.* |
| **void** | **addByteArray** (byte size, int dataOffset, byte[] data) throws Exception  
*Multiplayer Specific - Add a byte array to the current packet.* |
| **void** | **addInt** (int data) throws Exception  
*Multiplayer Specific - Add an int to the current packet.* |
| **void** | **addMultipleScoreEntry** (int score, int level, int scoreType)  
Adds one score for one level with no supplemental data to the multiple scores buffer for multiple scores sending. |
| **void** | **addMultipleScoreEntryWithSupplementalData** (int score, int level, int scoreType, int[] supplemental_data)  
Adds one score for one level with supplemental data to the multiple scores buffer for multiple scores sending. |
| **void** | **addShort** (short data) throws Exception  
*Multiplayer Specific - Add a short to the current packet.* |
| **void** | **addString** (String data) throws Exception  
*Multiplayer Specific - Add a string array to the current packet.* |
| **void** | **cancel** ()  
Cancels the ongoing request. |
| **void** | **cleanup** ()  
Cleans up the memory allocated during the last server request. |
| **void** | **clearData** ()  
*Multiplayer Specific - Clears the data for the current processing packet.* |
| **byte** | **getByte** () throws Exception  
*Multiplayer Specific - Get a byte from the current packet.* |
| **byte[]** | **getByteArray** (int dataOffset, byte[] datadest) throws Exception  
*Multiplayer Specific - Get a byte array from the current packet, with offset.* |
| **byte[]** | **getByteArray** () throws Exception  
*Multiplayer Specific - Get a byte array from the current packet.* |
| **byte[]** | **getByteDataList** ()  
*Multiplayer Specific - Get the list of specific byte data for the items hold in the internal buffers in certain multiplayer states.* |
| **int** | **getCurrentPlayerLeaderboardPosition** ()  
Gets this user's position from the currently accessed leaderboard. |
| **int** | **getCurrentPlayerLeaderboardScore** ()  
Gets this user's score from the currently accessed leaderboard. |
| **int[]** | **getCurrentPlayerLeaderboardScoreData** ()  
Gets this user's supplemental data from the currently accessed leaderboard. |
<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[]</td>
<td><strong>getData ()</strong></td>
<td>Multiplayer Specific - Gets the data for the current processing packet.</td>
</tr>
<tr>
<td>String[]</td>
<td><strong>getDataList ()</strong></td>
<td>Multiplayer Specific - Get the list of specific binary data for the items hold in the internal buffers in certain multiplayer states.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getFirstSessionIndex ()</strong></td>
<td>Multiplayer Specific - Get the index of the first session on the multiplayer server, retrieved by the list session command.</td>
</tr>
<tr>
<td>String</td>
<td><strong>getFoundPlayerName ()</strong></td>
<td>Multiplayer Specific - Get the name of the searched player, after a search operation has returned and was handeled with <code>mpHandleFindPlayer()</code>.</td>
</tr>
<tr>
<td>String</td>
<td><strong>getFoundPlayerSessionName ()</strong></td>
<td>Multiplayer Specific - Get the name of the session the searched player is in, if any, after a search operation has returned and was handeled with <code>mpHandleFindPlayer()</code>.</td>
</tr>
<tr>
<td>byte</td>
<td><strong>getFoundPlayerSessionNumberOfPlayers ()</strong></td>
<td>Multiplayer Specific - Get the number of players of the session the searched player is in, if any, after a search operation has returned and was handeled with <code>mpHandleFindPlayer()</code>.</td>
</tr>
<tr>
<td>byte</td>
<td><strong>getFoundPlayerStatus ()</strong></td>
<td>Multiplayer Specific - Get the status of the searched player, after a search operation has returned and was handeled with <code>mpHandleFindPlayer()</code>.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getInt ()</strong> throws Exception</td>
<td>Multiplayer Specific - Get an int from the current packet.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getLastError ()</strong></td>
<td>Gets the status of the last request.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>getLeaderboardData</strong> (String[] names, int[] positions, int[] scores, int[][] scoreDatas)</td>
<td>Get all the leaderboard data in a single request.</td>
</tr>
<tr>
<td>String</td>
<td><strong>getLeaderboardEntryPlayerName</strong> (int index)</td>
<td>Gets the name for the player at index in the server returned leaderboard player list.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getLeaderboardEntryPlayerPosition</strong> (int index)</td>
<td>Gets the ranking of the player at index in the server returned leaderboard player list.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getLeaderboardEntryPlayerScore</strong> (int index)</td>
<td>Gets the score of the player at index in the server returned leaderboard player list.</td>
</tr>
<tr>
<td>int[]</td>
<td><strong>getLeaderboardEntryPlayerScoreData</strong> (int index)</td>
<td>Gets the supplemental data for the player at index in the server returned leaderboard player list.</td>
</tr>
<tr>
<td>int</td>
<td><strong>getLeaderboardSize ()</strong></td>
<td>Gets the leaderboard size for the currently accessed leaderboard list.</td>
</tr>
</tbody>
</table>
int `getLength()`
- Multiplayer Specific - Get the data length for the current processing packet.

int `getMyAvgScore()`
- M7 Specific - Gets the user's average score.

int `getMyBestRank()`
- M7 Specific - Returns the user's highest ever rank.

int `getMyHighScore()`
- M7 Specific - Gets the user's best score.

int[] `getMyHighScoreData()`
- M7 Specific - Gets the user's highest score's supplemental data.

String `getMyLastTimePlayed()`
- M7 Specific - Gets the date and time the user last played.

int `getMyLowScore()`
- M7 Specific - Gets the user's lowest score.

int[] `getMyLowScoreData()`
- M7 Specific - Gets the user's lowest score's supplemental data.

int `getMyNumberOfGamesPlayed()`
- M7 Specific - Gets the number of times the user has played the game.

String[] `getNameList()`
- Multiplayer Specific - Get the names list of the items hold in the internal buffers in certain multiplayer states.

int `getNewRankAfterScoreSending()`
- Returns the player's new rank after a score is sent.

int `getNumberOfItems()`
- Multiplayer Specific - Get the number of sessions retrieved by the list session command.

void `getPlayerStats` (int[] bestRankArr, int[] highScoreArr, int[] lowScoreArr, int[] avgScoreArr, int[] numberOfGamesPlayedArr, String[] lastTimePlayedArr, int[] lowScoreDataArr, int[] highScoreDataArr)
- M7 Specific - Get all the player states in a single request, previously requested from the server using the `sendStatsGet` and `handleStatsGet` functions.

String `getRequestedPlayerData()`
- Multiplayer Specific - the data for the player whose specific data was requested using `mpSendGetPlayerData()` and `mpHandleGetPlayerData()`.

String `getRequestedPlayerNickname()`
- Multiplayer Specific - the nickname for the player whose specific data was requested using `mpSendGetPlayerData()` and `mpHandleGetPlayerData()`.

String `getSessionData()`
- Multiplayer Specific - Get the specific game data for the session the player is currently in.

String `getSessionName()`
- Multiplayer Specific - Get the name of the session the player is currently in.
<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| short    | `getShort()`                 | throws Exception
Multiplayer Specific - Get a short from the current packet.               |
| String   | `getString()`                | throws Exception
Multiplayer Specific - Get a string from the current packet.                |
<p>| String   | <code>getUsername()</code>              | Fetches the current username.                                               |
| void     | <code>handleChangeUsername()</code>     | Handles the login or registration requests, after such a request has been initiated with <code>sendChangeUsername()</code>, this function's asynchronous pair. |
| void     | <code>handleHighscore()</code>          | Handles the score uploading request, after such a request has been initiated with <code>sendHighscore()</code>, <code>sendHighscoreWithSupplementalData()</code> or <code>sendMultipleHighscores()</code>, this function's asynchronous pairs. |
| void     | <code>handleLogin()</code>              | Handles the login or registration requests, after such a request has been initiated with <code>sendLogin()</code>, this function's asynchronous pair. |
| void     | <code>handleRankGet()</code>            | Handles the leaderboard request, after such a request has been initiated with <code>sendRankGet()</code>, this function's asynchronous pair. |
| void     | <code>handleRankGetAroundPlayer()</code>| Handles the request for getting the leaderboard entries around the user's rank, after such a request has been initiated with <code>sendRankGetAroundPlayer()</code>, this function's asynchronous pair. |
| void     | <code>handleRateGame()</code>           | M7 Specific - Handles the rate game request, after such a request has been initiated with <code>sendRateGame()</code>, this function's asynchronous pair. |
| void     | <code>handleRecommendGame()</code>      | M7 Specific - Handles the recommend game request, after such a request has been initiated with <code>sendRecommendGame()</code>, this function's asynchronous pair. |
| void     | <code>handleStatsGet()</code>           | M7 Specific - Handles the request for the user's stats, after such a request has been initiated with <code>sendStatsGet()</code>, this function's asynchronous pair. |
| void     | <code>initMultipleScores()</code>       | Initialises the multiple scores buffer for multiple score sending.         |
| boolean  | <code>isLoggedIn()</code>               | Provides information on whether the user has logged or registered on the server yet. |
| void     | <code>mpDisconnect()</code>             | Multiplayer Specific - Finishes the connection with the multiplayer server. |
| void     | <code>mpHandleCreateSession()</code>    | Multiplayer Specific - Handles the create game session request, after such request has been initiated with <code>mpSendCreateSession</code>, this function's asynchronous pair. |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mpHandleDisconnect</strong> ()</td>
<td>Multiplayer Specific - Handles the disconnect request, after such request has been initiated with <code>mpSendDisconnect</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleEstablishConnection</strong> ()</td>
<td>Multiplayer Specific - Handles the connection establish request, after such request has been initiated with <code>mpSendEstablishConnection</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleFindPlayer</strong> ()</td>
<td>Multiplayer Specific - Handles the get player info request, after such request has been initiated with <code>mpSendFindPlayer()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleFinishGame</strong> ()</td>
<td>Multiplayer Specific - Handles the finish game request, after such request has been initiated with <code>mpSendFinishGame()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleGameData</strong> () throws Exception</td>
<td>Multiplayer Specific - receives a game data package, used for in game messages.</td>
</tr>
<tr>
<td><strong>mpHandleGetPlayerData</strong> ()</td>
<td>Multiplayer Specific - Handles the get player info request, after such request has been initiated with <code>mpSendGetPlayerData()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleJoinSession</strong> ()</td>
<td>Multiplayer Specific - Handles the join game session request, after such request has been initiated with <code>mpSendJoinSession</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleKickOutPlayer</strong> (String name)</td>
<td>Multiplayer Specific - Handles the kick out player request, after such request has been initiated with <code>mpSendKickOutPlayer()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleLeaveSession</strong> ()</td>
<td>Multiplayer Specific - Handles the leave game session request, after such request has been initiated with <code>mpSendLeaveSession</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleListSession</strong> ()</td>
<td>Multiplayer Specific - Handles the list session request, after such request has been initiated with <code>mpSendListSession</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleLogin</strong> ()</td>
<td>Multiplayer Specific - Handles the login request, after such request has been initiated with <code>mpSendLogin</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleStartGame</strong> ()</td>
<td>Multiplayer Specific - Handles the start game request, after such request has been initiated with <code>mpSendStartGame()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td><strong>mpHandleUpdates</strong> ()</td>
<td>Multiplayer Specific - Handles the messages received from server without a previous request - push and in game messages.</td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>final boolean mpHasOpponentFinished()</code></td>
<td>Multiplayer Specific - Test if the opponent has finished the game.</td>
</tr>
<tr>
<td><code>final boolean mpIsConnected()</code></td>
<td>Multiplayer Specific - Test if the player is connected to the multiplayer server.</td>
</tr>
<tr>
<td><code>final boolean mpIsInGame()</code></td>
<td>Multiplayer Specific - Test if the player is in a playing a game on the multiplayer server.</td>
</tr>
<tr>
<td><code>final boolean mpIsInSession()</code></td>
<td>Multiplayer Specific - Test if the player is in a game session on the multiplayer server.</td>
</tr>
<tr>
<td><code>final boolean mpIsLoggedIn()</code></td>
<td>Multiplayer Specific - Test if the player is logged on the multiplayer server.</td>
</tr>
<tr>
<td><code>final boolean mpIsMaster()</code></td>
<td>Multiplayer Specific - Test if the player is in a master in a session on the multiplayer server.</td>
</tr>
<tr>
<td><code>void mpPrepareGameData()</code></td>
<td>Multiplayer Specific - Initialize a game data package, used for in game messages.</td>
</tr>
<tr>
<td><code>void mpSendCreateSession(String sessionname, String sessiondata)</code></td>
<td>Multiplayer Specific - Create a new game on the multiplayer server.</td>
</tr>
<tr>
<td><code>void mpSendDisconnect()</code></td>
<td>Multiplayer Specific - Send a disconnect request to the multiplayer server.</td>
</tr>
<tr>
<td><code>void mpSendEstablishConnection()</code></td>
<td>Multiplayer Specific - Send an connection establish messages to the server.</td>
</tr>
<tr>
<td><code>void mpSendFindPlayer(String playerName)</code></td>
<td>Multiplayer Specific - get info for a player - status, current game.</td>
</tr>
<tr>
<td><code>void mpSendFinishGame()</code></td>
<td>Multiplayer Specific - Finish the game.</td>
</tr>
<tr>
<td><code>void mpSendGameData()</code></td>
<td>Multiplayer Specific - Send a game data package, used for in game messages.</td>
</tr>
<tr>
<td><code>void mpSendGetPlayerData(String playerName)</code></td>
<td>Multiplayer Specific - Get the player specific data for a specified player.</td>
</tr>
<tr>
<td><code>void mpSendJoinSession(String sessionname)</code></td>
<td>Multiplayer Specific - Join a game session on the multiplayer server.</td>
</tr>
<tr>
<td><code>void mpSendKickOutPlayer(String playerName)</code></td>
<td>Multiplayer Specific - Kick out a player.</td>
</tr>
<tr>
<td><code>void mpSendLeaveSession()</code></td>
<td>Multiplayer Specific - Leave a game session.</td>
</tr>
<tr>
<td><code>void mpSendListSession(byte numberOfSessions, byte firstSessionIndex)</code></td>
<td>Multiplayer Specific - Send a list available games request to the server.</td>
</tr>
<tr>
<td><code>void mpSendLogin(String player_data)</code></td>
<td>Multiplayer Specific - Send a login request to the server.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>void <strong>mpSendLogin</strong> ()</td>
<td>Multiplayer Specific - Send a login request to the server.</td>
</tr>
<tr>
<td>void <strong>mpSendQuickGame</strong> ()</td>
<td>Multiplayer Specific - Send a quick game request to the server.</td>
</tr>
<tr>
<td>void <strong>mpSendStartGame</strong> ()</td>
<td>Multiplayer Specific - Start the game.</td>
</tr>
<tr>
<td>void <strong>sendChangeUsername</strong> ()</td>
<td>Starts a request for changing username.</td>
</tr>
<tr>
<td>void <strong>sendHighscore</strong> (int score, int level, int scoreType)</td>
<td>Uploads one score for one level on the server, if the level has no supplemental data.</td>
</tr>
<tr>
<td>void <strong>sendHighscoreWithSupplementalData</strong> (int score, int level, int scoreType, int[] supplemental_data)</td>
<td>Uploads one score for one level on the server, with the supplemental data associated.</td>
</tr>
<tr>
<td>void <strong>sendLogin</strong> (String playerData)</td>
<td>Starts a request for either registration or login.</td>
</tr>
<tr>
<td>void <strong>sendMultipleHighscores</strong> ()</td>
<td>Uploads the scores stored in the multiple score buffer on the server.</td>
</tr>
<tr>
<td>void <strong>sendRankGet</strong> (int level, int number_of_players, int scoreType, int numberOfSupplementalDataFields)</td>
<td>Starts a request for one leaderboard.</td>
</tr>
<tr>
<td>void <strong>sendRankGetAroundPlayer</strong> (int level, int number_of_players_around, int scoreType, int numberOfSupplementalDataFields)</td>
<td>Starts a request for getting the scores of players ranked around the user's current standing in a leaderboard.</td>
</tr>
<tr>
<td>void <strong>sendRateGame</strong> (int rating)</td>
<td>M7 Specific - Sends a request for rating the game.</td>
</tr>
<tr>
<td>void <strong>sendRecommendGame</strong> ()</td>
<td>M7 Specific - Sends a request for recommending the game.</td>
</tr>
<tr>
<td>void <strong>sendStatsGet</strong> (int level, int numberOfSupplementalDataFields)</td>
<td>M7 Specific - Sends a request for the user's stats for a certain level.</td>
</tr>
<tr>
<td>void <strong>setData</strong> (byte[] data)</td>
<td>Multiplayer Specific - Sets the data for the current processing packet.</td>
</tr>
<tr>
<td>void <strong>setPhoneNumber</strong> (String phoneNr)</td>
<td>M7 Specific - Registers the phone number associated with the player's account.</td>
</tr>
<tr>
<td>void <strong>setUsername</strong> (String name)</td>
<td>Modifies the username.</td>
</tr>
</tbody>
</table>

**XPlayer** (MIDlet midlet)
Public Attributes

<table>
<thead>
<tr>
<th>boolean</th>
<th>isGameMessageInQueue = false</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Quick check if an IN_GAME message has been received.</td>
</tr>
</tbody>
</table>
Static Public Attributes

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static long</td>
<td>callstarttime</td>
<td>The time the current request started - used for timeout implementation.</td>
</tr>
<tr>
<td>static final int</td>
<td>NOT_A_NUMBER = -666666</td>
<td>Used to define anything that should be a number but it has not been initialised or does not have a value yet.</td>
</tr>
<tr>
<td>static final byte</td>
<td>SCORE_TYPE_POINTS = 1</td>
<td>Used to define scores that refer to points.</td>
</tr>
<tr>
<td>static final byte</td>
<td>SCORE_TYPE_TIME = 2</td>
<td>Used to define scores that are time-related.</td>
</tr>
</tbody>
</table>
### Classes

<table>
<thead>
<tr>
<th>interface</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ConnectLevel</td>
</tr>
<tr>
<td></td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>MessageType</td>
</tr>
</tbody>
</table>
Constructor & Destructor Documentation

XPlayer ( MIDlet midlet )

Constructor for the XPlayer class.

It reads the values of the XPlayerURL and GGI properties from the .jad file of midlet for initialisation purposes.

Parameters:

midlet  Reference to the MIDlet of the game which implements this API.
Member Function Documentation

void addByte ( byte data ) throws Exception

Multiplayer Specific - Add a byte to the current packet.

Parameters:
    data  The data to add

void addByteArray ( byte size, int dataOffset, byte[] data ) throws Exception

Multiplayer Specific - Add a byte array to the current packet.

Parameters:
    size  The size of data to add
    dataOffset  The offset in the input data
    data  The data to add

void addInt ( int data ) throws Exception

Multiplayer Specific - Add an int to the current packet.

Parameters:
    data  The data to add

void addMultipleScoreEntry ( int score, int level, int scoreType )

Adds one score for one level with no supplemental data to the multiple scores buffer for multiple scores sending.
This is used to build up a request for uploading multiple scores at once to the server using `sendMultipleHighscores()`.

Do not forget to initialise the buffer with `initMultipleScores()` before starting to add scores to it.

**Parameters:**

- **score**
  - The score to be uploaded.

- **level**
  - The level number for that score. If the game only has one leaderboard, don't send the level parameter! Call this function with level = -1.

- **scoreType**
  - Either `SCORE_TYPE_POINTS` or `SCORE_TYPE_TIME`, depending on the type of score.

**See also:**

- `initMultipleScores()`
- `addMultipleScoreEntryWithSupplementalData(int, int, int, int[])`
- `sendMultipleHighscores()`
- `SCORE_TYPE_POINTS`
- `SCORE_TYPE_TIME`

```c
void addMultipleScoreEntryWithSupplementalData ( int score,
                                 int level,
                                 int scoreType,
                                 int[] supplemental_data )
```

Adds one score for one level with supplemental data to the multiple scores buffer for multiple scores sending.

This is used to build up a request for uploading multiple scores at once to the server using `sendMultipleHighscores()`.

Do not forget to initialise the buffer with `initMultipleScores()` before starting to add scores to it.

**Parameters:**
**score**
The score to be uploaded.

**level**
The level number for that score. If the game only has one leaderboard, don't send the level parameter! Call this function with `level = -1`.

**scoreType**
Either `SCORE_TYPE_POINTS` or `SCORE_TYPE_TIME`, depending on the type of score.

**supplemental_data**
The supplemental data associated with the score to be sent.

See also:
- `initMultipleScores()`
- `addMultipleScoreEntry(int, int, int)`
- `sendMultipleHighscores()`
- `SCORE_TYPE_POINTS`
- `SCORE_TYPE_TIME`

```java
void addShort ( short data ) throws Exception
```
Multiplayer Specific - Add a short to the current packet.

**Parameters:**
- `data` The data to add

```java
void addString ( String data ) throws Exception
```
Multiplayer Specific - Add a string array to the current packet.

**Parameters:**
- `data` The data to add

```java
void cancel ( )
```
Cancels the ongoing request.
void cleanup ( )

Cleans up the memory allocated during the last server request.

Use this everytime a request completes (it finishes being pending - check with getLastError() to free up some memory.

See also:
getLastError()

void clearData ( )

Multiplayer Specific - Clears the data for the current processing packet.

See also:
setData(byte[])
getdata()

byte getByte ( ) throws Exception

Multiplayer Specific - Get a byte from the current packet.

Returns:
The data to get.

byte [] getByteArray ( int dataOffset, byte[] datadest )

throws Exception

Multiplayer Specific - Get a byte array from the current packet, with offset.

Parameters:

dataOffset  The offset from which the data will be copied in the result buffer.
datadest    The destination buffer to hold the data.
Returns:
The data to get.

```java
byte[] getByteArray() throws Exception
```

Multiplayer Specific - Get a byte array from the current packet.

Returns:
The data to get.

```java
byte[] getByteDataList()
```

Multiplayer Specific - Get the list of specific byte data for the items hold in the internal buffers in certain multiplayer states.

The items can be either sessions (if a list session request was performed) or players (if the current player is inside a game session). If the item is a session, this data is the number of players currently in session, if the item is a player, it is the index of the player in session.

Returns:
The list of data corresponding to each player.

```java
int getCurrentPlayerLeaderboardPosition()
```

Gets this user's position from the currently accessed leaderboard.

Note that for this function to return valid results:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The index must be less than the leaderboardSize.

Returns:
The user's rank within the most recently accessed leaderboard.

```
int getCurrentPlayerLeaderboardScore ( )
```

Gets this user's score from the currently accessed leaderboard.

Note that for this function to return a valid result:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The `index` must be less than the `leaderboardSize`.

**Returns:**
The user's score within the most recently accessed leaderboard.

```
int [] getCurrentPlayerLeaderboardScoreData ( )
```

Gets this user's supplemental data from the currently accessed leaderboard.

Note that for this function to return a valid result:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The `index` must be less than the `leaderboardSize`.

**Returns:**
The user's supplemental data within the most recently accessed leaderboard.

```
byte [] getData ( )
```
Multiplayer Specific - Gets the data for the current processing packet.

**Returns:**
The data to get.

**See also:**
`setData(byte[])`

---

**String [] getDataList ( )**

Multiplayer Specific - Get the list of specific binary data for the items hold in the internal buffers in certain multiplayer states.

The items can be either sessions, if a list session request was performed, or players, if the current player is inside a game session.

**Returns:**
The list of data corresponding to each player.

---

**int getFirstSessionIndex ( )**

Multiplayer Specific - Get the index of the first session on the multiplayer server, retrieved by the list session command.

**Returns:**
The index of the first session.

---

**String getFoundPlayerName ( )**

Multiplayer Specific - Get the name of the searched player, after a search operation has returned and was handeled with `mpHandleFindPlayer()`.

**Returns:**
The name of the searched player.
### String getFoundPlayerSessionName ( )

Multiplayer Specific - Get the name of the session the searched player is in, if any, after a search operation has returned and was handled with `mpHandleFindPlayer()`.

**Returns:**
- The name of the searched player.

### byte getFoundPlayerSessionNumberOfPlayers ( )

Multiplayer Specific - Get the number of players of the session the searched player is in, if any, after a search operation has returned and was handled with `mpHandleFindPlayer()`.

**Returns:**
- The name of the searched player.

### byte getFoundPlayerStatus ( )

Multiplayer Specific - Get the status of the searched player, after a search operation has returned and was handled with `mpHandleFindPlayer()`.

**Returns:**
- The status of the searched player.

### int getInt ( ) throws Exception

Multiplayer Specific - Get an int from the current packet.

**Returns:**
- The data to get.

### int getLastError ( )
Gets the status of the last request.

Call this after any handle function to find out the status of the request.

If ERROR_PENDING is returne, the call is still in progress.

**Returns:**
One of the error codes defined in the `XPlayer.Error` interface depending on the last request's status.

**See also:**
`XPlayer.Error`

```java
boolean getLeaderboardData ( String[] names,
                          int[] positions,
                          int[] scores,
                          int[] scoreDatas[][]
)
```

Get all the leaderboard data in a single request.

This function can be used instead of: `getLeaderboardEntryPlayerName`, `getLeaderboardEntryPlayerPosition`, `getLeaderboardEntryPlayerScore` and `getLeaderboardEntryPlayerScoreData`.

Please use only one way to get the leaderboard data, either this function or the single entry functions. That way the unused ones will be removed by the obfuscator.

The caller is responsable for allocating memory and parsing the returned array.

Note that for this function to return a valid result:

- A leaderboard request must have been successfully completed using `sendRankGet() / handleRankGet() / sendRankGetAroundPlayer() / handleRankGetAroundPlayer()` functions.

• The index must be is less then the leaderboardSize.

TODO: Add example of how to use this.

Parameters:

- names The names array to fill.
- positions The positions array to fill.
- scores The scores array to fill.
- scoreDatas The scode datas array to fill.

Returns:

true if leaderboard data was retrieved sucessfully.

See also:

- getLeaderboardEntryPlayerName(int)
- getLeaderboardEntryPlayerPosition(int)
- getLeaderboardEntryPlayerScore(int)
- getLeaderboardEntryPlayerScoreData(int)

String getLeaderboardEntryPlayerName ( int index )

Gets the name for the player at index in the server returned leaderboard player list.

Note that for this function to return valid results:

• A leaderboard request must have been successfully completed using sendRankGet() / handleRankGet() or sendRankGetAroundPlayer() / handleRankGetAroundPlayer() functions.
• The index must be is less then the leaderboardSize.

TODO: Add example of how the single entry functions are to be used.

Parameters:

- index The players index in the currently downloaded list to get the name for.
Returns:
The name of the player at $\text{index}$ within the leaderboard list or null if either $\text{index}$ exceeds the leaderboard size or there is no list previously downloaded.

```c
int getLeaderboardEntryPlayerPosition ( int index )
```

Gets the ranking of the player at $\text{index}$ in the server returned leaderboard player list.

Note that for this function to return valid results:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The $\text{index}$ must be is less then the leaderboardSize.

Parameters:

$\text{index}$ The players index in the currently downloaded list to get the rank for.

Returns:
The ranking for the player at $\text{index}$ within the leaderboard list or -1 if either $\text{index}$ exceeds the leaderboard size or there is no list previously downloaded.

```c
int getLeaderboardEntryPlayerScore ( int index )
```

Gets the score of the player at $\text{index}$ in the server returned leaderboard player list.

Note that for this function to return valid results:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The `index` must be less than the `leaderboardSize`.

**Parameters:**

`index` The players index in the currently downloaded list to get the score for.

**Returns:**

The ranking for the player at `index` within the leaderboard list or `NOT_A_NUMBER` if either `index` exceeds the leaderboard size or there is no list previously downloaded.

```java
int[] getLeaderboardEntryPlayerScoreData(int index)
```

Gets the supplemental data for the player at `index` in the server returned leaderboard player list.

Note that for this function to return valid results:

- A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.
- The `index` must be less than the `leaderboardSize`.

**Parameters:**

`index` The players index in the currently downloaded list to get the supplemental data for.

**Returns:**

The supplemental data for the player at `index` within the leaderboard list or `null` if either `index` exceeds the leaderboard size or there is no list previously downloaded.

```java
int getLeaderboardSize()
```

Gets the leaderboard size for the currently accessed leaderboard list.

Note that for this function to return valid results:
A leaderboard request must have been successfully completed using `sendRankGet()`/`handleRankGet()` or `sendRankGetAroundPlayer()`/`handleRankGetAroundPlayer()` functions.

- The index must be less than the leaderboardSize.

**Returns:**
The leaderboard size in number of players or -1 if the leaderboard is empty.

```java
int getLength ()
```

Multiplayer Specific - Get the data length for the current processing packet.

**Returns:**
The length of the package.

**See also:**
- `setData(byte[])`
- `getData()`

```java
int getMyAvgScore ()
```

M7 Specific - Gets the user's average score.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**
The average score for the user.

**See also:**
- `sendStatsGet(int, int)`
**int getMyBestRank ( )**

M7 Specific - Returns the user’s highest ever rank.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**

The best rank yet for the user.

**See also:**

`sendStatsGet(int, int)`

**int getMyHighScore ( )**

M7 Specific - Gets the user's best score.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**

The highest score ever for the user.

**See also:**

`sendStatsGet(int, int)`

**int [] getMyHighScoreData ( )**

M7 Specific - Gets the user's highest score's supplemental data.

This function is only valid for M7 Networks hosted games.
This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**
The user's highest score's supplemental data array or null if none.

**See also:**
`sendStatsGet(int, int)`

---

String `getMyLastTimePlayed ( )`

M7 Specific - Gets the date and time the user last played.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**
The date and time the user last played.

**See also:**
`sendStatsGet(int, int)`

---

Int `getMyLowScore ( )`

M7 Specific - Gets the user's lowest score.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**
The user's lowest score.

**See also:**
`sendStatsGet(int, int)`
### int [] getMyLowScoreData ( )

M7 Specific - Gets the user's lowest score's supplemental data.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**

The user's lowest score's supplemental data array or null if none.

**See also:**

`sendStatsGet(int, int)`

### int getMyNumberOfGamesPlayed ( )

M7 Specific - Gets the number of times the user has played the game.

This function is only valid for M7 Networks hosted games.

This function returns a valid result only if `sendStatsGet()` has been previously called.

**Returns:**

The number of times the user played.

**See also:**

`sendStatsGet(int, int)`

### String [] getNameList ( )

Multiplayer Specific - Get the names list of the items hold in the internal buffers in certain multiplayer states.

The items can be either sessions, if a list session request was
performed, or players, if the current player is inside a game session.

Returns:
The list of names.

```
int getNewRankAfterScoreSending()
```

Returns the player's new rank after a score is sent.

This function will work properly only if a score sending was performed before its call.

Returns:
The new rank.

```
int getNumberOfItems()
```

Multiplayer Specific - Get the number of sessions retrieved by the list session command.

Returns:
The number of sessions.

```
void getPlayerStats(int[] bestRankArr,
                   int[] highScoreArr,
                   int[] lowScoreArr,
                   int[] avgScoreArr,
                   int[] numberOfGamesPlayerdArr,
                   String[] lastTimePlayedArr,
                   int[] lowScoreDataArr,
                   int[] highScoreDataArr)
```

M7 Specific - Get all the player states in a single request, previously requested from the server using the sendStatsGet and handleStatsGet functions.
This function is only valid for M7 Networks hosted games.

This function can be used to save package size and memory instead of getMyHighScore, getMyLowScore, getMyAvgScore, getMyBestRank, getMyNumberOfGamesPlayed and getMyLastTimePlayed.

Please use only one way to get the player states, either this function or the single value functions. The unused ones will be removed by the obfuscator.

The parameters are passed as an array[1], to emulate C style "pass by reference" arguments.

**Important:**
The caller is responsible for allocating the arrays.

Note that for this function to return valid results, a get player states request must have been previously completed successfully using the above mentioned functions.

**Parameters:**

- `bestRankArr` Single value array to store bestRank.
- `highScoreArr` Single value array to store highScore.
- `lowScoreArr` Single value array to store lowScore.
- `avgScoreArr` Single value array to store avgScore.
- `numberOfGamesPlayedArr` Single value array to store numberOfGamePlayed.
- `lastTimePlayedArr` Single value array to store the lastTimePlayed.
- `lowScoreDataArr` Single value array to store the lowScore supplemental data.
- `highScoreDataArr` Single value array to store the highScore supplemental data.

**String getRequestedPlayerData()**

Multiplayer Specific - the data for the player whose specific data was requested using `mpSendGetPlayerData()` and `mpHandleGetPlayerData()`.

USE THIS IN CONJUNCTION WITH
ENABLE_PLAYER_SPECIFIC_DATA
**Returns:**
The specific player data.

**String getRequestedPlayerNickname ( )**

Multiplayer Specific - the nickname for the player whose specific data was requested using `mpSendGetPlayerData()` and `mpHandleGetPlayerData()`.

USE THIS IN CONJUNCTION WITH ENABLE PLAYER SPECIFIC DATA

**Returns:**
The nickname of the player.

**String getSessionData ( )**

Multiplayer Specific - Get the specific game data for the session the player is currently in.

**Returns:**
The session data.

**String getSessionName ( )**

Multiplayer Specific - Get the name of the session the player is currently in.

**Returns:**
The name of the session.

**short getShort ( ) throws Exception**

Multiplayer Specific - Get a short from the current packet.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Returns:</strong></td>
<td>The data to get.</td>
</tr>
<tr>
<td><strong>String getString () throws Exception</strong></td>
<td>Multiplayer Specific - Get a string from the current packet.</td>
</tr>
<tr>
<td><strong>Returns:</strong></td>
<td>The data to get.</td>
</tr>
<tr>
<td><strong>String getUsername ()</strong></td>
<td>Fetches the current username.</td>
</tr>
<tr>
<td><strong>Returns:</strong></td>
<td>The player's current username.</td>
</tr>
<tr>
<td><strong>See also:</strong></td>
<td>handleLogin()</td>
</tr>
<tr>
<td><strong>void handleChangeUsername ()</strong></td>
<td>Handles the login or registration requests, after such a request has been initiated with <code>sendChangeUsername()</code>, this function's asynchronous pair.</td>
</tr>
<tr>
<td></td>
<td>The status of the call can be checked with <code>getLastError()</code></td>
</tr>
<tr>
<td><strong>See also:</strong></td>
<td><code>sendChangeUsername()</code></td>
</tr>
<tr>
<td></td>
<td><code>getLastError()</code></td>
</tr>
<tr>
<td><strong>void handleHighscore ()</strong></td>
<td></td>
</tr>
</tbody>
</table>
Handles the score uploading request, after such a request has been initiated with either `sendHighscore()`, `sendHighscoreWithSupplementalData()` or `sendMultipleHighscores()`, this function's asynchronous pairs.

The status of the request can be checked with `getLastError()`.

See also:
- `sendHighscore(int, int, int)`
- `sendHighscoreWithSupplementalData(int, int, int[], int[])`
- `sendMultipleHighscores()`
- `getLastError()`

```java
void handleLogin ( )

Handles the login or registration requests, after such a request has been initiated with `sendLogin()`, this function's asynchronous pair.

The status of the call can be checked with `getLastError()`.

The login sequence has a particular call order to ensure every possible error is treated so below we give a detailed step-by-step description. Also, please check the example given in this class' description.

```
handleLogin();
_errCode = getLastError();

If the response is not pending anymore, something must have happened. Either an error or success, we move onto PROCESS to determine a course of action.

    if (_errCode != XPlayer.Error.ERROR_PENDING)
        {
            cleanup();
            setOnlineSubstate(OnlineSubstate.PROCESS);
        }
        break;
    }

    case OnlineSubstate.PROCESS:
        {
            switch (_errCode)
            {
                case XPlayer.Error.ERROR_NONE:
                    {
                        no error
                        success login - go to display
                        setOnlineSubstate(OnlineSubstate.DISPLAY);
                        break;
                    }

                case XPlayer.Error.ERROR_NICK_TAKEN:
                    {
                        The chose username is already taken so the user either types another one or he chooses the one the server automatically suggests. The suggestion is taken from the server response and copied as the current username so to get the suggested nickname we call:
                        suggestedName = getUsername();
                        Reset the nickname
                        setUsername('');
                        ...// Go to a state which informs the user the nickname is already taken and prompt him to type another one or choose the server-sent suggestion
                        break;
                    }

                case XPlayer.Error.ERROR_REGISTER_FAILED
                case XPlayer.Error.ERROR_NO_NICKNAME
                case XPlayer.Error.ERROR_NO_PHONE_NUMBER:
                    {
                        The user has not registered yet. He must be prompted to do
so and asked to type a username of his choice and his phone number.

First we make sure username and phone number is clean and ready for the user to edit

```java
    interactiv.setUsername(""enberg();
    interactiv.setPhoneNumber(""enberg();
```

...// Then we move to a state in which we ask the user to register. From that point on, if he chooses to register, he will have to be presented with the means to enter a username and his phone number.

```java
    break;
}
```

```java
    case XPlayer.Error.ERROR_CONNECTION
    {
        ...// A connection error occurred - for example inform the user of its appearance and then go to a new state. We should also provide the user the means to retry the request.
        break;
    }
```

```java
    default:
    {
        ...// Some other error occurred. Also, an error message should be displayed and an options to retry or quit provided.
        break;
    }
```

```java
} // End of switch (_errCode).
```

```java
    break; // Out of PROCESS.
```

```java
    case OnlineSubstate.DISPLAY:
    {
        ...// Display some welcome message or something to inform the successful login.
        break;
    }
```

See also:

```java
    sendLogin(String)
```

```java
    getLastError()
```
void handleRankGet( )

Handles the leaderboard request, after such a request has been initiated with `sendRankGet`, this function's asynchronous pair.

This function will check for a response or an error, update the error code accordingly (this can be checked with `getLastError`) and, if the request is successful, parse the message received from the server for the leaderboard data.

If the request finishes with a success:

- To get this user's ranking information use:
  1. `getCurrentPlayerLeaderboardPosition()`,
  2. `getCurrentPlayerLeaderboardScore()` and
  3. `getCurrentPlayerLeaderboardScoreData()`
- To get other players ranking information that has been requested use:
  1. `getLeaderboardSize()`,
  2. `getLeaderboardEntryPlayerName()`,
  3. `getLeaderboardEntryPlayerPosition()`,
  4. `getLeaderboardEntryPlayerScore()` and if needed
  5. `getLeaderboardEntryPlayerScoreData()`
  OR
  1. `getLeaderboardSize()` and
  2. `getLeaderboardData()`.

See also:
- `sendRankGet(int, int, int, int)`
- `getCurrentPlayerLeaderboardPosition()`
- `getCurrentPlayerLeaderboardScore()`
- `getCurrentPlayerLeaderboardScoreData()`
- `getLeaderboardSize()`
- `getLeaderboardEntryPlayerName(int)`
getLeaderboardEntryPlayerPosition(int)
getLeaderboardEntryPlayerScore(int)
getLeaderboardEntryPlayerScoreData(int)
getLeaderboardData(String[], int[], int[], int[][])

void handleRankGetAroundPlayer()

Handles the request for getting the leaderboard entries around the user's rank, after such a request has been initiated with sendRankGetAroundPlayer(), this function's asynchronous pair.

This function will check for a response or an error, update the error code accordingly (this can be checked with getLastError()) and, if the request is successful, parse the message received from the server for the leaderboard data.

If the request finishes with a success:

- To get this users ranking information use:
  1. get_current_player_leaderboard_position()
  2. get_current_player_leaderboard_score() and
  3. get_current_player_leaderboard_score_data()

- To get the other players ranking information that has been requested use
  1. get_leaderboard_size(),
  2. 
  3. get_leaderboard_entry_player_name(),
  4. get_leaderboard_entry_player_position(),
  5. get_leaderboard_entry_player_score() and if needed
  6. get_leaderboard_entry_player_score_data()

OR
  1. get_leaderboard_size() and
  2. get_leaderboard_data().

See also:
sendRankGetAroundPlayer(int, int, int, int)
getCurrentPlayerLeaderboardPosition()
getCurrentPlayerLeaderboardScore()
getCurrentPlayerLeaderboardScoreData()
getLeaderboardSize()
getLeaderboardEntryPlayerName(int)
getLeaderboardEntryPlayerPosition(int)
getLeaderboardEntryPlayerScore(int)
getLeaderboardEntryPlayerScoreData(int)
getLeaderboardData(String[], int[], int[], int[][])

void handleRateGame ( )

M7 Specific - Handles the rate game request, after such a request has been initiated with sendRateGame, this function's asynchronous pair.

Important:
Rating games is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

See also:  
  sendRateGame(int)

void handleRecommendGame ( )

M7 Specific - Handles the recommend game request, after such a request has been initiated with sendRecommendGame, this function's asynchronous pair.
Important:
Recommending games is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

See also:
sendRecommendGame()

void handleStatsGet ( )

M7 Specific - Handles the request for the user's stats, after such a request has been initiated with sendStatsGet(), this function's asynchronous pair.

Important:
User statistics is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

See also:
sendStatsGet(int, int)

void initMultipleScores ( )

Initialises the multiple scores buffer for multiple score sending.

Call this before you begin assembling a list of multiple scores to be sent in one request with sendMultipleHighscores(). You must then add scores to the buffer using either addMultipleScoreEntry() or addMultipleScoreEntryWithSupplementalData() before actually sending the request.

See also:
addMultipleScoreEntry(int, int, int)
addMultipleScoreEntryWithSupplementalData(int, int, int[])
### sendMultipleHighscores()

#### boolean isLoggedIn()

Provides information on whether the user has logged or registered on the server yet.

Returns `true` if login or registration has been initiated and has completed successfully and `false` otherwise.

The login operation must be performed only once per app started. If the login was successful, even if the user leaves the game lobby menu, when he returns the login will not be performed again.

**Returns:**

Returns `true` if user has already logged in onto the server or `false` otherwise.

---

### void mpDisconnect()

Multiplayer Specific - Finishes the connection with the multiplayer server.

---

### void mpHandleCreateSession()

Multiplayer Specific - Handles the create game session request, after such request has been initiated with `mpSendCreateSession`, this function's asynchronous pair.

**See also:**

`mpSendCreateSession(String, String)`

---

### void mpHandleDisconnect()

Multiplayer Specific - Handles the disconnect request, after such
request has been initiated with mpSendDisconnect, this function's asynchronous pair.

See also:
    mpHandleDisconnect()

void mpHandleEstablishConnection ( )

Multiplayer Specific - Handles the connection establish request, after such request has been initiated with mpSendEstablishConnection, this function's asynchronous pair.

Before any communication with the server takes place, the physical connection must be established.

See also:
    mpSendEstablishConnection()

void mpHandleFindPlayer( )

Multiplayer Specific - Handles the get player info request, after such request has been initiated with mpSendFindPlayer(), this function's asynchronous pair.

See also:
    mpSendFindPlayer(String)

void mpHandleFinishGame ( )

Multiplayer Specific - Handles the finish game request, after such request has been initiated with mpSendFinishGame(), this function's asynchronous pair.

See also:
    mpSendFinishGame()
void mpHandleGameData ( ) throws Exception

Multiplayer Specific - receives a game data package, used for in-game messages.

See also:
    mpSendGameData()

void mpHandleGetPlayerData ( )

Multiplayer Specific - Handles the get player info request, after such request has been initiated with mpSendGetPlayerData(), this function's asynchronous pair.

See also:
    mpSendGetPlayerData(String)

void mpHandleJoinSession ( )

Multiplayer Specific - Handles the join game session request, after such request has been initiated with mpSendJoinSession, this function's asynchronous pair.

When the response is received, it contains the session data and the list of players currently in session.

See also:
    mpSendJoinSession(String)

void mpHandleKickOutPlayer ( String name )

Multiplayer Specific - Handles the kick out player request, after such request has been initiated with mpSendKickOutPlayer(), this function's asynchronous pair.

Parameters:
**name**  The name of the player that has been requested to be kicked out.

**See also:**

mpSendKickOutPlayer(String)

**void mpHandleLeaveSession ( )**

Multiplayer Specific - Handles the leave game session request, after such request has been initiated with mpSendLeaveSession, this function's asynchronous pair.

**See also:**

mpSendLeaveSession()

**void mpHandleListSession ( )**

Multiplayer Specific - Handles the list session request, after such request has been initiated with mpSendListSession, this function's asynchronous pair.

**See also:**

mpSendListSession(byte,byte)

**void mpHandleLogin ( )**

Multiplayer Specific - Handles the login request, after such request has been initiated with mpSendLogin, this function's asynchronous pair.

**See also:**

mpSendLogin()

**void mpHandleStartGame ( )**

Multiplayer Specific - Handles the start game request, after such request has been initiated with mpSendStartGame(), this function's
asynchronous pair.

See also:

mpSendStartGame()

void mpHandleUpdates ( )

Multiplayer Specific - Handles the messages received from server without a previous request - push and in game messages.

This function must be called on every cycle when the player is in session or playing a game.

final boolean mpHasOpponentFinished ( )

Multiplayer Specific - Test if the opponent has finished the game.

Returns:

Returns true if finished, false otherwise.

final boolean mpIsConnected ( )

Multiplayer Specific - Test if the player is connected to the multiplayer server.

Returns:

Returns true if connected, false otherwise.

final boolean mpIsInGame ( )

Multiplayer Specific - Test if the player is in a playing a game on the multiplayer server.

Returns:

Returns true if in game, false otherwise.
### `final boolean mplsInSession()`

**Multiplayer Specific - Test** if the player is in a game session on the multiplayer server.

**Returns:**
Returns true if in session, false otherwise.

### `final boolean mplsLoggedIn()`

**Multiplayer Specific - Test** if the player is logged on the multiplayer server.

**Returns:**
Returns true if logged in, false otherwise.

### `final boolean mplsIsMaster()`

**Multiplayer Specific - Test** if the player is in a master in a session on the multiplayer server.

**Returns:**
Returns true if master, false otherwise.

### `void mpPrepareGameData()`

**Multiplayer Specific - Initialize** a game data package, used for in game messages.

**See also:**
`mpSendGameData()`

### `void mpSendCreateSession(String `sessionname`, String `sessiondata`)`
Multiplayer Specific - Create a new game on the multiplayer server.

The creator of the session is called master. He can kick out a player that joins the session or start the game after enough player have joined. The created game can be seen by all the players who perform a list session request. The joining of new players is signaled by a push message from the server. While being is session, the players list is available.

**Parameters:**

- `sessionname` The name of the session. This is identical with the player nickname.
- `sessiondata` Specific game data for the game session. This can include map, track, difficulty, etc.

**See also:**

`mpHandleCreateSession()`

---

```c
void mpSendDisconnect ( )
```

Multiplayer Specific - Send a disconnect request to the multiplayer server.

**See also:**

`mpHandleDisconnect()`

---

```c
void mpSendEstablishConnection ( )
```

Multiplayer Specific - Send an connection establish messages to the server.

**See also:**

`mpHandleEstablishConnection()`

---

```c
void mpSendFindPlayer ( String playerName )
```
Multiplayer Specific - get info for a player - status, current game.

Get the player specific data for a specified player.

See also:

mpHandleFindPlayer()

---

void mpSendFinishGame ( )

Multiplayer Specific - Finish the game.

This message is sent by each player after he finished playing the multiplayer game.

The player will be removed from the session. The difference between finish game and leave game is that the session is not terminated when the master finished the game, but waits for all the players to finish.

See also:

mpHandleFinishGame()

---

void mpSendGameData ( )

Multiplayer Specific - Send a game data package, used for in game messages.

See also:

mpPrepareGameData()

---

void mpSendGetPlayerData ( String playerName )

Multiplayer Specific - Get the player specific data for a specified player.

See also:

mpHandleGetPlayerData()
void mpSendJoinSession ( String sessionname )

Multiplayer Specific - Join a game session on the multiplayer server.

After the game is joined, the player will wait for the session master to start the game.

The players list is available while being in session.

**Parameters:**

*sessionname*  The name of the session to be joined.

**See also:**

mpHandleListSession()  

void mpSendKickOutPlayer ( String playerName )

Multiplayer Specific - Kick out a player.

The session master can kick out a player before the game is started. The kicked out player will be removed from the session and receive a push message informing them about it.

If a player that is not master tries to kick out another player, this will result in an error.

**Parameters:**

*playerName*  The name of the player to be kicked out.

**See also:**

mpHandleKickOutPlayer(String)  

void mpSendLeaveSession ( )

Multiplayer Specific - Leave a game session.

Every player can leave a session at any time. If the player is the...
master of the session, the session is terminated and all the other players will receive a kick out push message.

**See also:**

`mpHandleLeaveSession()`

```c
void mpSendListSession( byte numberOfSessions,
                       byte firstSessionIndex )
```

Multiplayer Specific - Send a list available games request to the server.

This is how it works: a request is made to the server, containing the number of sessions requested and the index of the first session. The server replies with the list, and the data is stored in internal buffers and can be accessed using the corresponding functions.

The sessions are retrieved by page, using the parameters.

For each session are returned the index of the session, the name, one data byte with the number of players currently in session and some binary data containing game specific session data (map, track, difficulty, etc.).

After the request is completed, the session list can be accessed with `getNameList`, `getDataList`, `getByteDataList`, `getNameList`, `getNumberOfItems` and `getFirstSessionIndex`.

**Parameters:**

- `numberOfSessions` The number of session requested.
- `firstSessionIndex` The index of the first session requested.

**See also:**

`mpHandleListSession()`

```c
void mpSendLogin( String player_data )
```

Multiplayer Specific - Send a login request to the server.
After the physical connection is established, the user must authenticate before using the server features.

The login is performed based on uid, no nickname is necessary.

This version of the function can be used to set the specific player data. Use this only in conjunction with ENABLE_PLAYER_SPECIFIC_DATA.

See also:

  mpHandleLogin()

void mpSendLogin ( )

Multiplayer Specific - Send a login request to the server.

After the physical connection is established, the user must authenticate before using the server features.

The login is performed based on uid, no nickname is necessary.

See also:

  mpHandleLogin()

void mpSendQuickGame ( )

Multiplayer Specific - Send a quick game request to the server.

The quick game request is used to get an available game from the server. The server responds either with a session list containing only one session, in which case a join request must be performed, either with no session, in which case a create session must be performed.

The handle function for this request is mpHandleListSession

See also:

  mpHandleListSession()
void mpSendStartGame ( )

Multiplayer Specific - Start the game.

The start signal can be sent only by the session master, after enough players have joined the session.

If another player sends the start signal or the master tries to start the game when there are not enough players yet, this will result in an error response.

After the game is started, the players can send in game messages to each other in whatever format they choose.

See also:  
mpHandleStartGame()

void sendChangeUsername ( )

Starts a request for changing username.

This function is asynchronous.

This request may be sent to the server to change the username. The status of the request will be checked using handleChangeUsername().

void sendHighscore ( int score,
                   int level,
                   int scoreType
                 )

Uploads one score for one level on the server, if the level has no supplemental data.

If the score for one level has one or more supplemental data fields, they must all be send in the request, otherwise the request will fail, so you must use the sendHighscoreWithSupplementalData() function.
If multiple scores are to be sent use `sendMultipleHighscores()`.

**Parameters:**

- **score**
  The score to be uploaded.
- **level**
  The level number for that score. If the game only has one leaderboard, don't send the level parameter! Call this function with level = -1.
- **scoreType**
  Either `SCORE_TYPE_POINTS` or `SCORE_TYPE_TIME`, depending on the type of score.

**See also:**

- `sendHighscoreWithSupplementalData(int, int, int, int[])`
- `SCORE_TYPE_POINTS`
- `SCORE_TYPE_TIME`
- `sendMultipleHighscores()`

```java
void sendHighscoreWithSupplementalData ( int score,
                                            int level,
                                            int scoreType,
                                            int[] supplemental_data )
```

Uploads one score for one level on the server, with the supplemental data associated.

Use this instead of `sendHighscore()` if the score to be sent has one or more supplemental data fields. Note that the supplemental data sent with the score must contain all fields required by the server (check your game's design documentation for the complete list) otherwise the request will fail.

If multiple scores are to be sent use `sendMultipleHighscores()`.

**Parameters:**

- **score**
  The score to be uploaded.
- **level**
  The level number for that score. If the game only has one leaderboard, don't send the level parameter! Call this function with level = -1.
scoreType Either SCORE_TYPE_POINTS or SCORE_TYPE_TIME, depending on the type of score.

supplemental_data The supplemental data associated with the score to be sent.

See also:

sendHighscore(int, int, int)

SCORE_TYPE_POINTS

SCORE_TYPE_TIME

sendMultipleHighscores()

void sendLogin(String playerData)

Starts a request for either registration or login.

This function is asynchronous.

The normal usage is sending login at the game load, after Gameloft logo, but before game splash screen and handle the response using handleLogin().

When the application is started, a login with empty user name and phone number must be performed. If the user is registered, the server will reply with the username. If not, a registration process is started. The username must not be saved on the device, but retrieved always from the server.

As this process involves a precise call order to ensure that every possible error is handled, a detailed description on how the login sequence should look is given in the handleLogin() function's definition.

Parameters:

playerData Is by default null. If the game has specific data associated with a player (like an avatar) it should be passed here.

See also:
handleLogin()

void sendMultipleHighscores ( )

Uploads the scores stored in the multiple score buffer on the server.

The buffer has to be initialised with initMultipleScores() and scores have to be added to it using either addMultipleScoreEntry() or addMultipleScoreEntryWithSupplementalData()

See also:
  initMultipleScores()
  addMultipleScoreEntry(int, int, int)
  addMultipleScoreEntryWithSupplementalData(int, int, int[], int)

void sendRankGet ( int level,
  int number_of_players,
  int scoreType,
  int numberOfSupplementalDataFields
)

Starts a request for one leaderboard.

This function is asynchronous.

Use handleRankGet() to handle this request.

Parameters:

  level
    The level to get the leaderboard for.
  number_of_players
    The maximum number of players in the leaderboard the server should return, starting with the highest ranked player.
  scoreType
    Either SCORE_TYPE_POINTS or SCORE_TYPE_TIME, depending on the type of score. It influences the way scores are ordered. If the scores are points, a higher score is better than a lower one. If the score is a time, a lower score is better than a higher one.
    - The number of supplemental data fields
  numberOfSupplementalDataFields

void sendRankGetAroundPlayer ( int level,
int number_of_players_around,
int scoreType,
int numberOfSupplementalDataFields
)

Starts a request for getting the scores of players ranked around the user's current standing in a leaderboard.

This function is asynchronous.

Use **handleRankGetAroundPlayer()** to handle this request.

**Parameters:**

- **level**
  The level to get the leaderboard for.

- **number_of_players_around**
  The maximum number of players that are around the user's rank in the leaderboard the server should return.

- **scoreType**
  Either SCORE_TYPE_POINTS or SCORE_TYPE_TIME, depending on the type of score. It influences the way scores are ordered. If the scores are points, a higher score is better than a lower one. If the score is a time, a lower score is better than a higher one.

- **numberOfSupplementalDataFields**
  - The number of supplemental data fields associated with this leaderboard. The CALLER is RESPONSABLE for providing this value correctly otherwise the request will fail!

**See also:**

**handleRankGetAroundPlayer()**

---

void sendRateGame ( int rating )

M7 Specific - Sends a request for rating the game.
This function is asynchronous and valid only on M7 Networks hosted games.

The valid ratings are 1-10. The description for each as stated in M7 official documentation is:

1. ROOT CANAL
2. GARBAGE
3. AWFUL
4. DISAPPOINTING
5. AVERAGE >
6. GOOD
7. GREAT
8. EXCELLENT
9. OUTSTANDING >
10. THE BEST

Use handleRateGame to handle the request afterwards.

**Important:**
Rating games is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

**Parameters:**

`rating` A number from 1 to 10, based on the rating the user chose for the game. See above.

**See also:**
handleRateGame()

```cpp
void sendRecommendGame ( )
```

M7 Specific - Sends a request for recommending the game.

This function is asynchronous and valid only for M7 Networks hosted games.

Use handleRecommendGame to handle the request afterwards.
**Important:**
Recommending games is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

**See also:**
`handleRecommendGame()`

```c
void sendStatsGet ( int level,
                   int numberOfSupplementalDataFields )
```

M7 Specific - Sends a request for the user's stats for a certain level.

This function is asynchronous and valid only on M7 Networks hosted games.

Use `handleStatsGet` to handle the request afterwards.

If the request completes successfully, use `getMyBestRank`, `getMyHighScore`, `getMyLowScore`, `getMyAvgScore`, `getMyNumberOfGamesPlayed`, `getMyLastTimePlayed` functions to retrieve the relevant data.

**Important:**
User statistics is a standard function for Sprint, Nextel, Cingular and Virgin Mobile (i.e. M7 Networks hosted games). This is NOT SUPPORTED for any other carriers.

**Parameters:**
- `level`
  - The level for which the stats are retrieved from the server.
  - The number of supplemental data fields associated with the specified level. The CALLER is RESPONSABLE for providing this value correctly otherwise the request will fail!

**See also:**
`handleStatsGet()`
void setData ( byte[] data )

Multiplayer Specific - Sets the data for the current processing packet.

Parameters:

  data  The data to set.

See also:

  getData()

void setPhoneNumber ( String phoneNr )

M7 Specific - Registers the phone number associated with the player's account.

Common practice is to ask the user for a username and his phone number upon registration and set them accordingly using this and the setUsername function.

Parameters:

  phoneNr  The string containing the phone number to be set.

See also:

  handleLogin()

  sendLogin(String)

void setUsername ( String name )

Modifies the username.

Common practice is to ask the user for a username and his phone number upon registration and set them accordingly using this and the setPhoneNumber() function.

The username must be set to "" in the initial registration check. The server will reply with the correct username. The only time the
**setUsername()** must be called with an actual username, is when the user is not yet registered with the server

**Parameters:**

- *name*  The string to be set as username.

**See also:**

- `handleLogin()`
### Member Data Documentation

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>long callstarttime</strong> [static]</td>
<td></td>
<td>The time the current request started - used for timeout implementation. Use only in conjunction with ENABLE_TIMEOUT.</td>
</tr>
<tr>
<td><strong>boolean isGameMessageInQueue = false</strong></td>
<td></td>
<td>Quick check if an IN_GAME message has been received. This is only used for in game messages!</td>
</tr>
<tr>
<td><strong>final int NOT_A_NUMBER = -666666 [static]</strong></td>
<td></td>
<td>Used to define anything that should be a number but it has not been initialised or does not have a value yet. For example when the user inquires about his score for a level that he has never played yet or anything similar.</td>
</tr>
<tr>
<td><strong>final byte SCORE_TYPE_POINTS = 1 [static]</strong></td>
<td></td>
<td>Used to define scores that refer to points. <strong>See also:</strong> <code>sendHighscore(int, int, int)</code></td>
</tr>
<tr>
<td><strong>final byte SCORE_TYPE_TIME = 2 [static]</strong></td>
<td></td>
<td>Used to define scores that are time-related. <strong>See also:</strong></td>
</tr>
</tbody>
</table>

---

**sendHighscore(int, int, int)**

A function for sending high scores to the server.
sendHighscore(int, int, int)
XPlayer.ConnectLevel
XPlayer.ConnectLevel Interface Reference

List of all members.
## Static Public Attributes

<table>
<thead>
<tr>
<th>static final byte</th>
<th>IN_CONNECTED = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>static final byte</td>
<td>IN_GAME = 5</td>
</tr>
<tr>
<td>static final byte</td>
<td>IN_GAME_MASTER = 6</td>
</tr>
<tr>
<td>static final byte</td>
<td>IN_SESSION = 3</td>
</tr>
<tr>
<td>static final byte</td>
<td>IN_SESSION_MASTER = 4</td>
</tr>
<tr>
<td>static final byte</td>
<td>NOT_CONNECTED = 0</td>
</tr>
<tr>
<td>static final byte</td>
<td>NOT_LOGGED_IN = 1</td>
</tr>
</tbody>
</table>
### Member Data Documentation

<table>
<thead>
<tr>
<th>Member</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>final byte IN_CONNECTED</td>
<td>2 [static]</td>
</tr>
<tr>
<td>final byte IN_GAME</td>
<td>5 [static]</td>
</tr>
<tr>
<td>final byte IN_GAME_MASTER</td>
<td>6 [static]</td>
</tr>
<tr>
<td>final byte IN_SESSION</td>
<td>3 [static]</td>
</tr>
<tr>
<td>final byte IN_SESSION_MASTER</td>
<td>4 [static]</td>
</tr>
<tr>
<td>final byte NOT_CONNECTED</td>
<td>0 [static]</td>
</tr>
<tr>
<td>final byte NOT_LOGGED_IN</td>
<td>1 [static]</td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:33 2008 for GLib by oxygen 1.5.2*
XPlayer.Error
XPlayer.Error Interface Reference

[Xplayer Error Code]

List of all members.
### Static Public Attributes

<table>
<thead>
<tr>
<th>Static final byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ERROR_BAD_RESPONSE = 40</code></td>
<td>The server response has incorrect format.</td>
</tr>
<tr>
<td><code>ERROR_CHANGE_USERNAME_FAILED = 33</code></td>
<td>Change username process failed.</td>
</tr>
<tr>
<td><code>ERROR_CONNECTION = -2</code></td>
<td>No connection to server.</td>
</tr>
<tr>
<td><code>ERROR_GET_RANKINGS_FAILED = 22</code></td>
<td>The get rankings operation failed for some reason.</td>
</tr>
<tr>
<td><code>ERROR_GET_STATS_FAILED = 29</code></td>
<td>Failed to get the states for the player.</td>
</tr>
<tr>
<td><code>ERROR_INIT = -100</code></td>
<td>Initialisation value.</td>
</tr>
<tr>
<td><code>ERROR_INVALID_GGI = 27</code></td>
<td>Configuration error: Most likely an invalid GGI.</td>
</tr>
<tr>
<td><code>ERROR_JOIN_GAME = 53</code></td>
<td>Multiplayer error - there was an error joining the game.</td>
</tr>
<tr>
<td><code>ERROR_NICK_TAKEN = 5</code></td>
<td>This nickname is taken, another one is suggested.</td>
</tr>
<tr>
<td><code>ERROR_NO_CLIENT_ID = 26</code></td>
<td>Programming error on emulator: no phone id.</td>
</tr>
<tr>
<td><code>ERROR_NO_NICKNAME = 2</code></td>
<td>The nickname is missing from the registration request.</td>
</tr>
<tr>
<td><code>ERROR_NO_PHONE_NUMBER = 25</code></td>
<td>There is no phone number in the register request.</td>
</tr>
<tr>
<td><code>ERROR_NO_UUID = 1</code></td>
<td>The uuid is missing from the request.</td>
</tr>
<tr>
<td><code>ERROR_NONE = 0</code></td>
<td>The request finished successfully.</td>
</tr>
<tr>
<td><code>ERROR_NOT_M7_ENABLED = -3</code></td>
<td>No connection to server.</td>
</tr>
<tr>
<td><code>ERROR_NOT_REGISTERED = 4</code></td>
<td>This user is not registered.</td>
</tr>
<tr>
<td><code>ERROR_PENDING = -1</code></td>
<td>Result is pending, the transaction with the server is ongoing.</td>
</tr>
<tr>
<td><code>ERROR_RATE_GAME_FAILED = 23</code></td>
<td>Failed to rate this game.</td>
</tr>
<tr>
<td><code>ERROR_RECOMMEND_GAME_FAILED = 24</code></td>
<td>Failed to recommend this game.</td>
</tr>
<tr>
<td>Static final byte</td>
<td>Code Name</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>ERROR_REGISTER_FAILED = 3</td>
<td></td>
</tr>
<tr>
<td>ERROR_REQUEST_FAILED = 54</td>
<td></td>
</tr>
<tr>
<td>ERROR_SCORE_UPLOAD_FAILED = 21</td>
<td></td>
</tr>
<tr>
<td>ERROR_START_GAME = 52</td>
<td></td>
</tr>
<tr>
<td>ERROR_SUPPLEMENTAL_DATA_NEEDED = 31</td>
<td></td>
</tr>
<tr>
<td>ERROR_VALIDATE_LICENSE_FAILED = 32</td>
<td></td>
</tr>
<tr>
<td>ERROR_WRONGFULL_QSTATE = 50</td>
<td></td>
</tr>
<tr>
<td>ERROR_WRONGFULL_RSTATE = 51</td>
<td></td>
</tr>
</tbody>
</table>
# Member Data Documentation

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>The server response has incorrect format. Make sure the request is correct. If it is a leaderboard request, make sure the number of supplemental data sent match the number for the leaderboard.</td>
</tr>
<tr>
<td>33</td>
<td>Change username process failed.</td>
</tr>
<tr>
<td>-2</td>
<td>No connection to server. Either there is no network available or the server cannot be contacted for some reason.</td>
</tr>
<tr>
<td>22</td>
<td>The get rankings operation failed for some reason.</td>
</tr>
<tr>
<td>29</td>
<td>Failed to get the states for the player.</td>
</tr>
<tr>
<td>-100</td>
<td>Initialisation value.</td>
</tr>
</tbody>
</table>
Getting this error means that actually no requests have been made yet since the MIDlet started so there is no last error.

final byte ERROR_INVALID_GGI = 27 [static]

Configuration error: Most likely an invalid GGI.

final byte ERROR_JOIN_GAME = 53 [static]

Multiplayer error - there was an error joining the game.

final byte ERROR_NICK_TAKEN = 5 [static]

This nickname is taken, another one is suggested.

final byte ERROR_NO_CLIENT_ID = 26 [static]

Programming error on emulator: no phone id.

final byte ERROR_NO_NICKNAME = 2 [static]

The nickname is missing from the registration request.

final byte ERROR_NO_PHONE_NUMBER = 25 [static]

There is no phone number in the register request.

final byte ERROR_NO_UUID = 1 [static]

The uuid is missing from the request.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERROR_NONE</strong></td>
<td>final byte <code>ERROR_NONE = 0</code> [static]</td>
</tr>
<tr>
<td></td>
<td>The request finished successfully.</td>
</tr>
<tr>
<td><strong>ERROR_NOT_M7_ENABLED</strong></td>
<td>final byte <code>ERROR_NOT_M7_ENABLED = -3</code> [static]</td>
</tr>
<tr>
<td></td>
<td>No connection to server.</td>
</tr>
<tr>
<td></td>
<td>Either there is no network available or the server cannot be contacted for</td>
</tr>
<tr>
<td></td>
<td>some reason.</td>
</tr>
<tr>
<td><strong>ERROR_NOT_REGISTERED</strong></td>
<td>final byte <code>ERROR_NOT_REGISTERED = 4</code> [static]</td>
</tr>
<tr>
<td></td>
<td>This user is not registered.</td>
</tr>
<tr>
<td><strong>ERROR_PENDING</strong></td>
<td>final byte <code>ERROR_PENDING = -1</code> [static]</td>
</tr>
<tr>
<td></td>
<td>Result is pending, the transaction with the server is ongoing.</td>
</tr>
<tr>
<td><strong>ERROR_RATE_GAME_FAILED</strong></td>
<td>final byte <code>ERROR_RATE_GAME_FAILED = 23</code> [static]</td>
</tr>
<tr>
<td></td>
<td>Failed to rate this game.</td>
</tr>
<tr>
<td><strong>ERROR_RECOMMEND_GAME_FAILED</strong></td>
<td>final byte <code>ERROR_RECOMMEND_GAME_FAILED = 24</code> [static]</td>
</tr>
<tr>
<td></td>
<td>Failed to recommend this game.</td>
</tr>
<tr>
<td><strong>ERROR_REGISTER_FAILED</strong></td>
<td>final byte <code>ERROR_REGISTER_FAILED = 3</code> [static]</td>
</tr>
<tr>
<td></td>
<td>The registration process failed.</td>
</tr>
</tbody>
</table>
final byte **ERROR_REQUEST_FAILED** = 54 [static]

Multiplayer error - there was a server error processing the request.

final byte **ERROR_SCORE_UPLOAD_FAILED** = 21 [static]

The score was not posted for some reason.

final byte **ERROR_START_GAME** = 52 [static]

Multiplayer error - there was an error starting the game.

final byte **ERROR_SUPPLEMENTAL_DATA_NEEDED** = 31 [static]

Programming error: the supplemental data needed to upload this leaderboard was not correctly supplied.

final byte **ERROR_VALIDATE_LICENSE_FAILED** = 32 [static]

Cingular MRC **License** Validation failed.

final byte **ERROR_WRONGFULL_QSTATE** = 50 [static]

Multiplayer error - the call has no relevance in this state.

The game and **XPlayer** think they're in different states.

final byte **ERROR_WRONGFULL_RSTATE** = 51 [static]

Multiplayer error - this response is not expected.
The game and XPlayer think they're in different states.
XPlayer::MessageType
XPlayer.MessageType Interface Reference

List of all members.
<table>
<thead>
<tr>
<th>Static final byte</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BASIC_MESSAGE_TYPE_CONNECT</strong> = 's'</td>
<td>Connect message type</td>
</tr>
<tr>
<td><strong>BASIC_MESSAGE_TYPE_GAME</strong> = 'g'</td>
<td>Game message type</td>
</tr>
<tr>
<td><strong>BASIC_MESSAGE_TYPE_UNKNOWN</strong> = 0</td>
<td>Unknown message type</td>
</tr>
<tr>
<td><strong>CONNECT_MESSAGE_DISCONNECT</strong> = 'x'</td>
<td>Disconnect message</td>
</tr>
<tr>
<td><strong>CONNECT_MESSAGE_ERROR</strong> = 'e'</td>
<td>Error message for connect</td>
</tr>
<tr>
<td><strong>CONNECT_MESSAGE_INIT_READ</strong> = 'r'</td>
<td>Init read message</td>
</tr>
<tr>
<td><strong>CONNECT_MESSAGE_INIT_WRITE</strong> = 'w'</td>
<td>Init write message</td>
</tr>
<tr>
<td><strong>CONNECT_MESSAGE_SUCCESS</strong> = 's'</td>
<td>Success message for connect</td>
</tr>
<tr>
<td><strong>ERROR_BLOCKED_SESSION</strong> = 'b'</td>
<td>Error for blocked session</td>
</tr>
<tr>
<td><strong>ERROR_CREATE_SESSION_INVALID_NAME</strong> = 'v'</td>
<td>Error for invalid session name</td>
</tr>
<tr>
<td><strong>ERROR_CREATE_SESSION_NAME_USED</strong> = 'u'</td>
<td>Error for session name already used</td>
</tr>
<tr>
<td><strong>ERROR_INVALID_INPUT</strong> = 'i'</td>
<td>Error for invalid input</td>
</tr>
<tr>
<td><strong>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</strong> = 'j'</td>
<td>Error for too many players to join session</td>
</tr>
<tr>
<td><strong>ERROR_KICK_OUT_PLAYER_IS_MASTER</strong> = 'd'</td>
<td>Error for kicking out player who is master</td>
</tr>
<tr>
<td><strong>ERROR_LIST_SESSION_INVALID_INDEX</strong> = 'l'</td>
<td>Error for invalid session index</td>
</tr>
<tr>
<td><strong>ERROR_LOGIN_AUTHENTICATION_FAILED</strong> = 'a'</td>
<td>Error for login authentication failed</td>
</tr>
<tr>
<td><strong>ERROR_LOGIN_INVALID_NICKNAME</strong> = 'q'</td>
<td>Error for invalid nickname login</td>
</tr>
<tr>
<td><strong>ERROR_LOGIN_NICKNAME_USED</strong> = 'n'</td>
<td>Error for nickname already used</td>
</tr>
<tr>
<td><strong>ERROR_NO_PLAYER</strong> = 'k'</td>
<td>Error for no player</td>
</tr>
<tr>
<td><strong>ERROR_NO_SESSION</strong> = 's'</td>
<td>Error for no session</td>
</tr>
<tr>
<td><strong>ERROR_NOT_MASTER</strong> = 'm'</td>
<td>Error for not master</td>
</tr>
<tr>
<td><strong>ERROR_SESSION_CLOSED</strong> = 'c'</td>
<td>Error for session closed</td>
</tr>
<tr>
<td><strong>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</strong> = 'g'</td>
<td>Error for not enough players to start game</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_ERROR</strong> = 'e'</td>
<td>Error message for game</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_IN_GAME</strong> = 'g'</td>
<td>In game message</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_IN_GAME_TOALL</strong> = 'h'</td>
<td>In game to all message</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_KEEP_ALIVE</strong> = 'a'</td>
<td>Keep alive message</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_PUSH</strong> = 'p'</td>
<td>Push message</td>
</tr>
<tr>
<td><strong>GAME_MESSAGE_REQUEST</strong> = 'r'</td>
<td>Request message</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_HAS_NO_PSD</strong> = 'd'</td>
<td>Player status has no PSD</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_HAS_PSD</strong> = 'h'</td>
<td>Player status has PSD</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_NOT_REGISTERED</strong> = 'n'</td>
<td>Player status not registered</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_OFFLINE</strong> = 'f'</td>
<td>Player status offline</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_ONLINE</strong> = 'o'</td>
<td>Player status online</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_ONLINE_IN_SESSION</strong> = 's'</td>
<td>Player status online in session</td>
</tr>
<tr>
<td><strong>PLAYER_STATUS_ONLINE_PLAYING</strong> = 'p'</td>
<td>Player status online playing</td>
</tr>
<tr>
<td><strong>PUSH_MESSAGE_FINISH_GAME</strong> = 'f'</td>
<td>Push message for finish game</td>
</tr>
<tr>
<td><strong>PUSH_MESSAGE_JOIN_SESSION</strong> = 'j'</td>
<td>Push message for join session</td>
</tr>
<tr>
<td><strong>PUSH_MESSAGE_KICK_OUT</strong> = 'k'</td>
<td>Push message for kick out</td>
</tr>
</tbody>
</table>

**Static Public Attributes**
<table>
<thead>
<tr>
<th>Static final byte</th>
<th>Message</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUSH_MESSAGE_LEAVE_SESSION</td>
<td>= 'l'</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_START_GAME</td>
<td>= 's'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_CREATE_SESSION</td>
<td>= 'c'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_FINISH_GAME</td>
<td>= 'f'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_DATA</td>
<td>= 'y'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_INFO</td>
<td>= 'w'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_QUICK_SESSION</td>
<td>= 'u'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_JOIN_SESSION</td>
<td>= 'j'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_KICK_OUT_PLAYER</td>
<td>= 'k'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LEAVE_SESSION</td>
<td>= 'q'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LIST_SESSIONS</td>
<td>= 'l'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LOGIN</td>
<td>= 'i'</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_START_GAME</td>
<td>= 's'</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_ERROR</td>
<td>= 'e'</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_SUCCESS</td>
<td>= 's'</td>
<td></td>
</tr>
</tbody>
</table>
### Member Data Documentation

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>final byte BASIC_MESSAGE_TYPE_CONNECT = 's'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte BASIC_MESSAGE_TYPE_GAME = 'g'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte BASIC_MESSAGE_TYPE_UNKNOWN = 0</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte CONNECT_MESSAGE_DISCONNECT = 'x'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte CONNECT_MESSAGE_ERROR = 'e'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte CONNECT_MESSAGE_INIT_READ = 'r'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte CONNECT_MESSAGE_INIT_WRITE = 'w'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte CONNECT_MESSAGE_SUCCESS = 's'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte ERROR_BLOCKED_SESSION = 'b'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte ERROR_CREATE_SESSION_INVALID_NAME = 'v'</code></td>
<td>[static]</td>
</tr>
<tr>
<td><code>final byte ERROR_CREATE_SESSION_NAME_USED = 'u'</code></td>
<td>[static]</td>
</tr>
<tr>
<td>Error Code</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ERROR_INVALID_INPUT</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_KICK_OUT_PLAYER_IS_MASTER</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_LIST_SESSION_INVALID_INDEX</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_LOGIN_AUTHENTICATION_FAILED</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_LOGIN_INVALID_NICKNAME</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_LOGIN_NICKNAME_USED</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_NO_PLAYER</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_NO_SESSION</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_NOT_MASTER</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_SESSION_CLOSED</td>
<td>final byte</td>
</tr>
<tr>
<td>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</td>
<td>final byte</td>
</tr>
</tbody>
</table>
final byte GAME_MESSAGE_ERROR = 'e' [static]

final byte GAME_MESSAGE_IN_GAME = 'g' [static]

final byte GAME_MESSAGE_IN_GAME_TOALL = 'h' [static]

final byte GAME_MESSAGE_KEEP_ALIVE = 'a' [static]

final byte GAME_MESSAGE_PUSH = 'p' [static]

final byte GAME_MESSAGE_REQUEST = 'r' [static]

final byte PLAYER_STATUS_HAS_NO_PSD = 'd' [static]

final byte PLAYER_STATUS_HAS_PSD = 'h' [static]

final byte PLAYER_STATUS_NOT_REGISTERED = 'n' [static]

final byte PLAYER_STATUS_OFFLINE = 'f' [static]

final byte PLAYER_STATUS_ONLINE = 'o' [static]

final byte PLAYER_STATUS_ONLINE_IN_SESSION = 's' [static]
final byte PLAYER_STATUS_ONLINE_PLAYING = 'p' [static]

final byte PUSH_MESSAGE_FINISH_GAME = 'f' [static]

final byte PUSH_MESSAGE_JOIN_SESSION = 'j' [static]

final byte PUSH_MESSAGE_KICK_OUT = 'k' [static]

final byte PUSH_MESSAGE_LEAVE_SESSION = 'l' [static]

final byte PUSH_MESSAGE_START_GAME = 's' [static]

final byte REQUEST_MESSAGE_CREATE_SESSION = 'c' [static]

final byte REQUEST_MESSAGE_FINISH_GAME = 'f' [static]

final byte REQUEST_MESSAGE_GET_PLAYER_DATA = 'y' [static]

final byte REQUEST_MESSAGE_GET_PLAYER_INFO = 'w' [static]

final byte REQUEST_MESSAGE_GET_QUICK_SESSION = 'u' [static]

final byte REQUEST_MESSAGE_JOIN_SESSION = 'j' [static]
<table>
<thead>
<tr>
<th>final byte REQUEST_MESSAGE_KICK_OUT_PLAYER = 'k' [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td>final byte REQUEST_MESSAGE_LEAVE_SESSION = 'q' [static]</td>
</tr>
<tr>
<td>final byte REQUEST_MESSAGE_LIST_SESSIONS = 'l' [static]</td>
</tr>
<tr>
<td>final byte REQUEST_MESSAGE_LOGIN = 'i' [static]</td>
</tr>
<tr>
<td>final byte REQUEST_MESSAGE_START_GAME = 's' [static]</td>
</tr>
<tr>
<td>final byte RESPONSE_MESSAGE_ERROR = 'e' [static]</td>
</tr>
<tr>
<td>final byte RESPONSE_MESSAGE_SUCCESS = 's' [static]</td>
</tr>
</tbody>
</table>
GLLib Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- ASprite
- Canvas
  - GLLib
- GLKey
- GLLang
- GLLibConfig
- GLLibMidiSpectrumAnalyzer
- GLLibPathFinding
- GRPH
- License
- License.Error
- MessageType
- Runnable
  - GLLib
  - GLLibPlayer
  - HTTP
- TCP
- XPlayer
- XPlayer.ConnectLevel
- XPlayer.Error
- XPlayer.MessageType
Here is a list of all class members with links to the classes they belong to:

- [ ]

  - [static initializer]() : ASprite , GLLib
GLLib Package List

Here are the packages with brief descriptions (if available):

<table>
<thead>
<tr>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.io</td>
</tr>
<tr>
<td>java.util</td>
</tr>
<tr>
<td>javax.microedition.io</td>
</tr>
<tr>
<td>javax.microedition.lcdui</td>
</tr>
<tr>
<td>javax.microedition.midlet</td>
</tr>
<tr>
<td>javax.microedition.rms</td>
</tr>
</tbody>
</table>
Package javax.microedition.lcdui

Generated on Tue Sep 23 23:05:33 2008 for GLib by doxygen 1.5.2
Package javax.microedition.rms
Here is a list of all files with brief descriptions:

<table>
<thead>
<tr>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASprite.java</td>
</tr>
<tr>
<td>BBKey.java</td>
</tr>
<tr>
<td>DummyPacketSender.java</td>
</tr>
<tr>
<td>GLLib.java</td>
</tr>
<tr>
<td>GLLibConfig.java</td>
</tr>
<tr>
<td>GLLibInterface.java</td>
</tr>
<tr>
<td>GLLibMidiSpectrumAnalyzer.java</td>
</tr>
<tr>
<td>GLLibPathFinding.java</td>
</tr>
<tr>
<td>GLLibPlayer.java</td>
</tr>
<tr>
<td>GloftBBUiApp.java</td>
</tr>
<tr>
<td>Graphics.java</td>
</tr>
<tr>
<td>GRPH.java</td>
</tr>
<tr>
<td>HTTP.java</td>
</tr>
<tr>
<td>Image.java</td>
</tr>
<tr>
<td>License.java</td>
</tr>
<tr>
<td>MessageType.java</td>
</tr>
<tr>
<td>RecordStore.java</td>
</tr>
<tr>
<td>TCP.java</td>
</tr>
<tr>
<td>XPlayer.java</td>
</tr>
</tbody>
</table>
ASprite.java File Reference
Namespaces

| namespace | javax.microedition.lcdui |
## Classes

<table>
<thead>
<tr>
<th>class</th>
<th>ASprite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASprite Aurora Sprite class. More...</td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:30 2008 for GLib by doxygen 1.5.2*
### Namespaces

<table>
<thead>
<tr>
<th>namespace</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.io</td>
<td>namespace</td>
</tr>
<tr>
<td>java.util</td>
<td>namespace</td>
</tr>
<tr>
<td>javax.microedition.rms</td>
<td>namespace</td>
</tr>
</tbody>
</table>
Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLib</th>
</tr>
</thead>
</table>

The GLLib class is the main class to do a game creation at Gameloft. [More...](#)
GLLibConfig.java File Reference
## Classes

| class       | GLLibConfig
|-------------|-------------
|             | *GLLib setup class.* [More...](#) |
GLLibInterface.java File Reference
## Classes

<table>
<thead>
<tr>
<th>interface</th>
<th>GLKey</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Game key code interface. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>interface</th>
<th>GLLang</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>language code <a href="#">More...</a></td>
</tr>
</tbody>
</table>
GLLibMidiSpectrumAnalyzer.java File Reference
<table>
<thead>
<tr>
<th>class</th>
<th>GLLibMidiSpectrumAnalyzer</th>
</tr>
</thead>
</table>

**Author:**
Nikolay Aleksiev

More...
### Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLibPathFinding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A* (A star) Pathfinding class. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:30 2008 for GLLib by [doxygen](http://www.stackoverflow.com) 1.5.2
GLLibPlayer.java File Reference
Classes

<table>
<thead>
<tr>
<th>class</th>
<th>GLLibPlayer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Generic GLLib Player. More...</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:30 2008 for GLLib by doxygen 1.5.2
GRPH.java File Reference
## Classes

<table>
<thead>
<tr>
<th>interface</th>
<th>GRPH</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATileSet Aurora TileSet class.</td>
<td>More...</td>
</tr>
</tbody>
</table>

*Generated on Tue Sep 23 23:05:30 2008 for GLLib by doxygen 1.5.2*
HTTP.java File Reference
Namespaces

| namespace | javax.microedition.io |
### Classes

| class | HTTP |

*Generated on Tue Sep 23 23:05:30 2008 for GLib by doxygen 1.5.2*
License.java File Reference
**Namespaces**

| namespace | javax.microedition.midlet |
## Classes

<table>
<thead>
<tr>
<th>class</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The base class for License communication with the Gameloft server. <a href="#">More...</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>interface</th>
<th>License.Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>License error codes. <a href="#">More...</a></td>
</tr>
</tbody>
</table>
MessageType.java File Reference
Classes

interface MessageType

Generated on Tue Sep 23 23:05:30 2008 for GLib by doxygen 1.5.2
TCP.java File Reference
Classes

class TCP
XPlayer.java File Reference
## Classes

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>XPlayer</td>
</tr>
<tr>
<td>interface</td>
<td>XPlayer.ConnectLevel</td>
</tr>
<tr>
<td>interface</td>
<td>XPlayer.Error</td>
</tr>
<tr>
<td>interface</td>
<td>XPlayer.MessageType</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:30 2008 for GLLib by **doxygen** 1.5.2
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
</table>

GLLib Related Pages

Here is a list of all related documentation pages:

- Deprecated List
Deprecated List

Member `GLLibPlayer.SetAnim` (int anim)
    use SetAnim (int anim, int nbLoop) instead

Member `GLLibPlayer.Update`()
    use void Update(int DT) instead

Member `GLLibPlayer.Tileset_GetCamera` (int nLayer)
    use Tileset_GetCameraX and Tileset_GetCameraY instead

Member `GLLibPlayer.Tileset_LoadLayer` (int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, boolean bUseCB, int origin, int wrappingX, int wrappingY)
    use Tileset_LoadLayer(int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, int iUseCB, int origin, int wrappingX, int wrappingY) instead

Member `GLLibPlayer.Tileset_Update` (int nLayer)
    do not use anymore

---

Generated on Tue Sep 23 23:05:30 2008 for GLlib by Doxygen 1.5.2
### GLLib Class Index

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>G</th>
<th>H</th>
<th>L</th>
<th>M</th>
<th>R</th>
<th>T</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASprite</td>
<td>GLLang</td>
<td>GRPH</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLLibConfig</td>
<td>GLLib</td>
<td>handleMessageType</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canvas (javax.microedition.lcdui)</td>
<td>GLLibMidiSpectrumAnalyzer</td>
<td>HTTP</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLKey</td>
<td>GLLibPathFinding</td>
<td>License</td>
<td>TCPLicense::Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GLLibPlayer</td>
<td>License::Error</td>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
# ASprite Member List

This is the complete list of members for *ASprite*, including all inherited members.

```plaintext
<table>
<thead>
<tr>
<th>[static initializer]</th>
<th>ASprite [package, static]</th>
</tr>
</thead>
<tbody>
<tr>
<td>_aframes</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_flags</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_frame</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_ox_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_ox_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_oy_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_oy_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aframes_time</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_alpha</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_anims_af_start</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_anims_naf</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedFlags</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedImages</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedImgX</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedImgY</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedSizeX</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedSizeY</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedX</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_aryPrecomputedY</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_bs_flags</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_bTraceNow</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_buffer_index</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_colors</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_cur_pal</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_cur_pool</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_data_format</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_flags</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_id</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_ox_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_ox_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_oy_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_oy_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_fmodules_pal</td>
<td>ASprite [package]</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Variable/Constant</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>_frames_col</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_col_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_fm_start</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_nfm</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_rc</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_rc_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_rects</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_rects_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_frames_rects_start</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_gifHeader</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_graphics</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_header_size</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>i64rle_color_bits</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>i64rle_color_mask</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_images_count</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_images_size</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_index1</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_index2</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_itoa_buffer</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>_main_image</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_map</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_colors_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_colors_int</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_image_imageAA</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_image_imageAAA</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_image_intAAA</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_module_types</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_data</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_data_off_int</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_data_off_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_extra_info</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_extra_pointer</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_h_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_h_scaled</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_h_short</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_image_shortAAA</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_usage</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_w_byte</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>_modules_w_scaled</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>Symbol</td>
<td>Type</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>_modules_w_short</td>
<td>ASprite</td>
</tr>
<tr>
<td>_modules_x_byte</td>
<td>ASprite</td>
</tr>
<tr>
<td>_modules_x_short</td>
<td>ASprite</td>
</tr>
<tr>
<td>modules_y_byte</td>
<td>ASprite</td>
</tr>
<tr>
<td>modules_y_short</td>
<td>ASprite</td>
</tr>
<tr>
<td>_nModules</td>
<td>ASprite</td>
</tr>
<tr>
<td>_old_pal</td>
<td>ASprite</td>
</tr>
<tr>
<td>_operation</td>
<td>ASprite</td>
</tr>
<tr>
<td>_pal_data</td>
<td>ASprite</td>
</tr>
<tr>
<td>_pal_int</td>
<td>ASprite</td>
</tr>
<tr>
<td>_pal_short</td>
<td>ASprite</td>
</tr>
<tr>
<td>_palettes</td>
<td>ASprite</td>
</tr>
<tr>
<td>_png_index</td>
<td>ASprite</td>
</tr>
<tr>
<td>_PNG_packed_IDAT_ADLER</td>
<td>ASprite</td>
</tr>
<tr>
<td>_PNG_packed_IDAT_CRC</td>
<td>ASprite</td>
</tr>
<tr>
<td>_PNG_packed_IHDR_CRC</td>
<td>ASprite</td>
</tr>
<tr>
<td>_PNG_packed_PLTE_CRC</td>
<td>ASprite</td>
</tr>
<tr>
<td>_PNG_packed_tRNS_CRC</td>
<td>ASprite</td>
</tr>
<tr>
<td>_png_result</td>
<td>ASprite</td>
</tr>
<tr>
<td>_png_size</td>
<td>ASprite</td>
</tr>
<tr>
<td>_png_start_crc</td>
<td>ASprite</td>
</tr>
<tr>
<td>_poolCacheSprites</td>
<td>ASprite</td>
</tr>
<tr>
<td>_poolCacheStack</td>
<td>ASprite</td>
</tr>
<tr>
<td>_poolCacheStackIndex</td>
<td>ASprite</td>
</tr>
<tr>
<td>_poolCacheStackMax</td>
<td>ASprite</td>
</tr>
<tr>
<td>_rectX1</td>
<td>ASprite</td>
</tr>
<tr>
<td>_rectX2</td>
<td>ASprite</td>
</tr>
<tr>
<td>_rectY1</td>
<td>ASprite</td>
</tr>
<tr>
<td>_rectY2</td>
<td>ASprite</td>
</tr>
<tr>
<td>_text_h</td>
<td>ASprite</td>
</tr>
<tr>
<td>_text_w</td>
<td>ASprite</td>
</tr>
<tr>
<td>_transp</td>
<td>ASprite</td>
</tr>
<tr>
<td>_w_pos</td>
<td>ASprite</td>
</tr>
<tr>
<td>_warpTextInfo</td>
<td>ASprite</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>ASprite()</td>
<td></td>
</tr>
<tr>
<td>BASE</td>
<td></td>
</tr>
<tr>
<td>BLOCK_INFO_SIZE</td>
<td>[package]</td>
</tr>
<tr>
<td>BS_AF_OFF_SHORT</td>
<td>[static]</td>
</tr>
<tr>
<td>BS_ANIMS</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_DOJA</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_MIDP1</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_MIDP1b</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_MIDP1c</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_MIDP2</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_DEFAULT_NOKIA</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_FM_OFF_SHORT</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_FM_PALETTE</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_FRAME_COLL_RC</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_FRAME_RECTS</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_FRAMES</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_GIF_HEADER</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_IMAGE_SIZE_INT</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_KEEP_PAL</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULE_IMAGES</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULE_IMAGES_FX</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULE_USAGE</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES_IMG</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES_USAGE</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES_WH_SHORT</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES_XY</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_MODULES_XY_SHORT</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BS_NAF_1_BYTE</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BS_NFM_1_BYTE</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_NO_AF_START</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_PNG_CRC</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_SINGLE_IMAGE</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_SKIP_FRAME_RC</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_TRANSP_FIRST</td>
<td>Static class</td>
</tr>
<tr>
<td>BS_TRANSP_LAST</td>
<td>Static class</td>
</tr>
<tr>
<td>BSPRITE_v003</td>
<td>Static class</td>
</tr>
<tr>
<td>BSPRITE_v004</td>
<td>Static class</td>
</tr>
<tr>
<td>BSPRITE_v005</td>
<td>Static class</td>
</tr>
<tr>
<td>BuildAnimCacheImages</td>
<td>Build cache images</td>
</tr>
<tr>
<td>BuildCacheImages</td>
<td>Build cache images</td>
</tr>
<tr>
<td>BuildFrameCacheImages</td>
<td>Build frame cache images</td>
</tr>
<tr>
<td>CountFrameModules</td>
<td>Count frame modules</td>
</tr>
<tr>
<td>Crc32</td>
<td>Crc32</td>
</tr>
<tr>
<td>crcTable</td>
<td>Crc32</td>
</tr>
<tr>
<td>currentChunkType</td>
<td>Current chunk type</td>
</tr>
<tr>
<td>DecodeImage</td>
<td>Decode image</td>
</tr>
<tr>
<td>DecodeImageToByteArray</td>
<td>Decode image to byte array</td>
</tr>
<tr>
<td>DrawLine</td>
<td>Draw line</td>
</tr>
<tr>
<td>DrawNumber</td>
<td>Draw number</td>
</tr>
<tr>
<td>DrawPage</td>
<td>Draw page</td>
</tr>
<tr>
<td>DrawPageB</td>
<td>Draw page with bounding box</td>
</tr>
<tr>
<td>DrawString</td>
<td>Draw string</td>
</tr>
<tr>
<td>DrawStringOrChars</td>
<td>Draw string or characters</td>
</tr>
<tr>
<td>Crc32 POLYNOMIAL</td>
<td>Constant polynomial</td>
</tr>
<tr>
<td>crcTable</td>
<td>Constant table</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>anchor, boolean restorecol</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I127RLE</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I16</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I12</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I256</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I256RLE</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I4</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>ENCODE_FORMAT_I64RLE</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_FLIP_X</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_FLIP_Y</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_HYPER_FM</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_INDEX_EX_MASK</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_OFFSET_AF</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_OFFSET_FM</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_ROT_90</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_USER0</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FLAG_USER1</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>FreeAnimCacheImages(int palette, int anim)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>FreeCacheData()</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>FreeFrameCacheImages(int palette, int frame)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>FreeMemory()</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>FreeModuleImage(int nPal, int nMod)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GenPalette(int type, int[] pal)</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>GenPalette(int type, short[] pal)</td>
<td>ASprite [package, static]</td>
</tr>
<tr>
<td>GetAFrameFlags(int anim, int aframe)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAFrameRect(int[] rc, int anim, int aframe, int posX, int posY, int flags)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAFrameRect(int anim, int aframe, int rectIndex, int[] rect, int flags)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAFrames(int anim)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAFramesOX(int v)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAFramesOY(int v)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAnimationFrame(int anim, int aframe)</td>
<td>ASprite [package]</td>
</tr>
<tr>
<td>GetAnimFrame(int anim, int aframe)</td>
<td>ASprite [package]</td>
</tr>
</tbody>
</table>
GetBold()  ASprite [package]
GetCharFrame(int c)  ASprite [package]
GetCharMap()  ASprite [package]
GetCharMapShort()  ASprite [package]
GetChars(char[] charBuf, int i, int radix, int minDigit)  ASprite [package, static]
getCharSize(char c)  ASprite [package]
GetCharSpacing()  ASprite [package]
GetCurrentMMapping()  ASprite [package]
GetCurrentPalette()  ASprite [package]
GetCurrentStringLength()  ASprite [package, static]
GetCurrentStringLengthWidth()  ASprite [package, static]
GetFModuleOx(int v)  ASprite [package]
GetFModuleOy(int v)  ASprite [package]
GetFModuleRect(int[] rc, int frame, int fmodule, int posX, int posY, int flags, int hx, int hy)  ASprite [package]
GetFModules(int frame)  ASprite [package]
GetFontHeight()  ASprite [package]
GetFrameCount()  ASprite [package]
GetFrameHeight(int frame)  ASprite [package]
GetFrameMarkers(int frame)  ASprite [package]
GetFrameModule(int frame, int fmodule)  ASprite [package]
GetFrameModuleFlags(int frame, int fmodule)  ASprite [package]
GetFrameModuleHeight(int frame, int fmodule)  ASprite [package]
GetFrameModulePalette(int frame, int fmodule)  ASprite [package]
GetFrameModuleWidth(int frame, int fmodule)  ASprite [package]
GetFrameModuleX(int frame, int fmodule)  ASprite [package]
GetFrameModuleY(int frame, int fmodule)  ASprite [package]
GetFrameRect(int[] rc, int frame, int posX, int posY, int flags, int hx, int hy)  ASprite [package]
GetFrameRect(int[] rc, int frame, int posX, int posY, int flags)  ASprite [package]
GetFrameRect(int frame, int rectIndex, int[] rect, int flags)  ASprite [package]
GetFrameRectCount(int frame)  ASprite [package]
GetFrames()  ASprite [package]
GetFrameWidth(int frame)  ASprite [package]
GetLineHeight()  ASprite [package]
GetLineSpacing()  ASprite [package]
GetModuleCount()  ASprite [package]
GetModuleData(int nModule, int nPalette)  ASprite [package]
GetModuleHeight(int module)  ASprite [package]
GetModuleHeightOrg(int module)  ASprite [package]
GetModuleImage(int nModule, int nPalette)  ASprite [package]
GetModuleRect(int[] rc, int module, int posX, int posY, int flags)  ASprite [package]
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
<th>Package</th>
<th>Static</th>
<th>Protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetModuleWidth</td>
<td>int module</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetModuleWidthOrg</td>
<td>int module</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetModuleX</td>
<td>int module</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetModuleY</td>
<td>int module</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetPalette</td>
<td>int nPalette</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetSpaceWidth</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GetUnderline</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEADER_LEVEL0_MAX_WBITS</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hRef</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hTarget</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDAT</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEND</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IHDR</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_EX_MASK</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_EX_SHIFT</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INDEX_MASK</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INFO32</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INFO8</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InitCachePool</td>
<td>int poolCount</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InitPoolSize</td>
<td>int poolIndex, int size</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>k_itoa_buffer_size</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>byte[] file, int offset</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>byte[] file, int offset, int pal_flags, int tr_flags</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>byte[] file, int offset, Image sprImage</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>byte[] file, int offset, int pal_flags, int tr_flags, Image sprImage</td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGIC</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGIC_IDAT_h</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGIC_IEND</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAX_TRANSFORMATION_FLAGS</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_ARC</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_FILL_ARC</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_FILL_RECT</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_FILL_TRIANGLE</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEADER_LEVEL0_MAX_WBITS</td>
<td></td>
<td>ASprite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symbol</td>
<td>Documentation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_IMAGE</td>
<td>ASprite static</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_MARKER</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_RECT</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MD_TRIANGLES</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mem</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>midp2_flags</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mod</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModifyPalette(int palNb, int color)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModifyPaletteAlpha(int palNb, int alpha)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModifyPaletteAlphaUsingAltPalette(int p_iPaletteID, int p_iAlphaPaletteID)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ModifyPaletteAlphaUsingLastPalette(int p_iPaletteID)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mResizeCorrectY</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mResizeRef</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NMAX</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATION_COMPUTERECT</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATION_DRAW</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATION_MARK</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPERATION_RECORD</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintAFrame(Graphics g, int anim, int aframe, int posX, int posY, int flags)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintAFrame(Graphics g, int anim, int aframe, int posX, int posY, int flags, int hx, int hy)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintFModule(Graphics g, int frame, int fmodule, int posX, int posY, int flags)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintFModule(Graphics g, int frame, int fmodule, int posX, int posY, int flags, int hx, int hy)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintFrame(Graphics g, int frame, int posX, int posY, int flags)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintFrame(Graphics g, int frame, int posX, int posY, int flags, int hx, int hy)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintModule(Graphics g, int module, int posX, int posY, int flags)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PaintPrecomputedFrame(Graphics g, int x, int y, int frame)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_BLEND Black</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_BLUE_CYAN</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_GREEN</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_GREY</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_INVISIBLE</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_ORIGINAL</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAL_RED_YELLOW</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIXEL_FORMAT_0565</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIXEL_FORMAT_1555</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIXEL_FORMAT_4444</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIXEL_FORMAT_8888</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLTE</td>
<td>ASprite [protected, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNG_INFO_SIZE</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrecomputeAllFrames(Graphics g)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PrecomputeFrame(Graphics g, int frame, int flags)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>record_frame</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>record_index</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResetCachePool(int poolIndex)</td>
<td>ASprite [static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_CreaterGB</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_DRAW_ON_MUTABLE</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_NONE</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_NOT_CACHED</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_REF_176x220</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_REF_240x256</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESIZE_REF_240x320</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ResizeCoords()</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_bAspectRatio</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_bBilinear</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_MapChar</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_rc</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s_resizeType</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCALE_SHIFT</td>
<td>ASprite [package, static]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scaleX(int x)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>scaleY(int y)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetBold(boolean bBold)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCharMap(byte[] pNewMap)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCharMap(short[] pNewMap)</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method/Function</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCharMapStatic</td>
<td>static byte[] pNewMap</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCharSpacing</td>
<td>int spacing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCharSpacingToDefault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCurrentMMapping</td>
<td>int map</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetCurrentPalette</td>
<td>int pal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetDefaultFontMetrics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetGraphics</td>
<td>Graphics g1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetLineHeight</td>
<td>int nHeight</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetLineHeightToDefault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetLineSpacing</td>
<td>int spacing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetLineSpacingToDefault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetModuleImage</td>
<td>Image plmg, int nModule, int nPalette</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetModuleImagesArray</td>
<td>Object pData</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetModuleMapping</td>
<td>int map, byte[] mmp</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetPool</td>
<td>int poolIndex</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetResizeParameters</td>
<td>int spriteId, int resizeMode, boolean correctY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetSpaceWidth</td>
<td>int spacing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetSpaceWidthToDefault</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetSubString</td>
<td>int i1, int i2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetTempBuffer</td>
<td>Object pData</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SetUnderline</td>
<td>boolean bUnderline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StringTokenize</td>
<td>String s, int index1, int index2, char token, int[] indices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUPPORTED_VERSION</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp_byte</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp_int</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>temp_short</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TextFitToFixedWidth</td>
<td>byte[] str, int text_w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TextFitToFixedWidth</td>
<td>String str, int text_w</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transform_int</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransformRGB</td>
<td>int[] image_data, int sizeX, int sizeY, int flags</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tRNS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unload</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateNumberSize</td>
<td>int num, int radix, int minDigit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateStringOrCharsSize</td>
<td>String s, char[] charBuff</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateStringSize</td>
<td>byte[] bs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UpdateStringSize</td>
<td>String s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WraptextB</td>
<td>String s, int width, int height</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wRef</td>
<td>ASprite</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wTarget</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xRatio</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yRatio</td>
<td>ASprite [package]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:32 2008 for GLLib by doxygen 1.5.2
GLKey Member List

This is the complete list of members for **GLKey**, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>k_consoleLogEnable</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_down</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_dummy</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_fire</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_invalid</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_left</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_menuBack</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_menuOK</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_nbKey</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num0</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num1</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num2</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num3</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num4</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num5</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num6</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num7</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num8</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_num9</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_pound</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_right</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_rmsLogEnable</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_star</td>
<td>GLKey</td>
</tr>
<tr>
<td>k_up</td>
<td>GLKey</td>
</tr>
</tbody>
</table>
This is the complete list of members for **GLLang**, including all inherited members.

<table>
<thead>
<tr>
<th>Language</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>CN</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>CZ</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>DE</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>EN</td>
<td>GLLang [package, static]</td>
</tr>
<tr>
<td>ES</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>FR</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>IT</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>JP</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>KR</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>PL</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>PT</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>RU</td>
<td>GLLang [static]</td>
</tr>
<tr>
<td>TR</td>
<td>GLLang [static]</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:32 2008 for GLLib by [doxygen](https://www.doxygen.nl) 1.5.2
# GLLib Member List

This is the complete list of members for **GLLib**, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Package</th>
<th>Static</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[static initializer]</strong></td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Align</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Assert</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>BOTTOM</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>CheckAndDumpConfig()</td>
<td>GLLib</td>
<td>[package]</td>
</tr>
<tr>
<td>ClipRect(int x, int y, int width, int height)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>ConvertFixedPointToString(int value, int precision)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>CopyArea(int x_src, int y_src, int width, int height, int x_dest, int y_dest, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Dbg(String log)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DOTTED</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawArc(int x, int y, int width, int height, int startAngle, int arcAngle)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawChar(char character, int x, int y, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawChars(char[] data, int offset, int length, int x, int y, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawImage(javax.microedition.lcdui.Image img, int x, int y, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawLine(int x1, int y1, int x2, int y2)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>drawPartialRGB(Graphics g, int rgbData[], int scanlength, int srcX, int srcY, int x, int y, int width, int height, boolean processAlpha)</td>
<td>GLLib</td>
<td>[static]</td>
</tr>
<tr>
<td>drawPartialRGB(Graphics g, int screenWidth, int screenHeight, int rgbData[], int scanlength, int srcX, int srcY, int x, int y, int width, int height, boolean processAlpha)</td>
<td>GLLib</td>
<td>[static]</td>
</tr>
<tr>
<td>DrawRect(int x, int y, int width, int height)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawRegion(javax.microedition.lcdui.Image img, int x_src, int y_src, int width, int height, int transform, int x_dest, int y_dest, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawRGB(int[] rgbData, int offset, int scanlength, int x, int y, int width, int height, boolean processAlpha)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawString(String str, int x, int y, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>DrawSubstring(String str, int offset, int len, int x, int y, int anchor)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Function</td>
<td>Package</td>
<td>Access</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>FillArc(int x, int y, int width, int height, int startAngle, int arcAngle)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>FillRect(int x, int y, int width, int height)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>FillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>FillTriangle(int x1, int y1, int x2, int y2, int x3, int y3)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>g</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>Game_KeyClearKeyCode()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>Game_KeySetKeyCode(boolean gameAction, int keyCode, int key)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>Game_Run()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Game_update()</td>
<td>GLLib</td>
<td>[package, pure virtual]</td>
</tr>
<tr>
<td>Gc()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetBlueComponent()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetClipHeight()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetClipWidth()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetClipX()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetClipY()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetColor()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetDisplayColor(int color)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetFont()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetFrameTime()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetGrayScale()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetGreenComponent()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetMIME(int idx)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetNBData()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetRealTime()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetRedComponent()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetResourceAsStream(String s)</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetScreenHeight()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetScreenWidth()</td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>GetSoftwareDoubleBuffer()</td>
<td>GLLib</td>
<td>protected</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>GetSoftwareDoubleBufferGraphics()</td>
<td>GLLib [protected]</td>
<td></td>
</tr>
<tr>
<td>GetStrokeStyle()</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>GetTranslateX()</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>GetTranslateY()</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>GLLib(Object application, Object display)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>HCENTER</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>hideNotify()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Init()</td>
<td>GLLib [protected]</td>
<td></td>
</tr>
<tr>
<td>InitSharedRms(String strVendor, String strMidletName)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>inputIndex</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsAnyKeyDown()</td>
<td>GLLib [static]</td>
<td></td>
</tr>
<tr>
<td>IsKeyDown(int keyFlag)</td>
<td>GLLib [static]</td>
<td></td>
</tr>
<tr>
<td>IsKeyUp(int keyFlag)</td>
<td>GLLib [static]</td>
<td></td>
</tr>
<tr>
<td>IsMatch</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsRep</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsRep0Long</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsRepG0</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsRepG1</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>IsRepG2</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kAlignTableSize</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kBitModelTotal</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kEndPosModelIndex</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>keyPressed(int keyCode)</td>
<td>GLLib [protected]</td>
<td></td>
</tr>
<tr>
<td>keyReleased(int keyCode)</td>
<td>GLLib [protected]</td>
<td></td>
</tr>
<tr>
<td>kLenNumHighBits</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kLenNumHighSymbols</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kLenNumLowBits</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kLenNumLowSymbols</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kLenNumMidBits</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kLenNumMidSymbols</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>kMatchMinLen</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td>Type</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>kNumAlignBits</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumBitModelTotalBits</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumFullDistances</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumLenProbs</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumLenToPosStates</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumMoveBits</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumPosBitsMax</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumPosSlotBits</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumPosStatesMax</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumStates</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kNumTopBits</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kStartPosModelIndex v</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>kTopValue</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LEFT</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenChoice</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenChoice2</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenCoder</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenHigh</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenLow</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LenMid</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Literal</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LZMA_BASE_SIZE</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LZMA_Inflate(byte[] compressDat)</td>
<td>GLLib</td>
<td>[static]</td>
</tr>
<tr>
<td>LZMA_LIT_SIZE</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LZMA_RESULT_DATA_ERROR</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LZMA_RESULT_NOT_ENOUGH_MEM</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>LZMA_RESULT_OK</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>m_Buffer</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>m_Code</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_current_keys_pressed</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_current_keys_released</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_current_keys_state</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_ExtraBytes</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_inSize</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_keys_pressed</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_keys_released</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_keys_state</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_last_key_pressed</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_outStream</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>m_Range</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Abs(int a)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Abs(long a)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Angle180</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Angle270</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Angle360</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Angle90</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_AngleMUL</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Atan(int dx, int dy)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_AtanSlow(int dx, int dy)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Bezier2D(int x1, int y1, int x2, int y2, int x3, int y3, int interp)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Bezier3D(int x1, int y1, int z1, int x2, int y2, int z2, int x3, int y3, int z3, int interp)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Cos(int angle)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_DegreeToFixedPointAngle(int a)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Det(int x1, int y1, int x2, int y2)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_DistPointLine(int x0, int y0, int x1, int y1, int x2, int y2)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Div10(int a)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Math_Div10(int a)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Package</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Math_DotProduct(int x1, int y1, int x2, int y2)</td>
<td>Calculates the dot product of two 2D vectors</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Add(int summand1, int summand2)</td>
<td>Adds two integers</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_DegreesToAngleFixedPoint(int angle)</td>
<td>Converts degrees to angle fixed point</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_DegreesToRadians(int angle)</td>
<td>Converts degrees to radians</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Det(int x1, int y1, int x2, int y2)</td>
<td>Calculates the determinant of a 2x2 matrix</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Divide(int dividend, int divisor)</td>
<td>Divides one integer by another integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_DotProduct(int x1, int y1, int x2, int y2)</td>
<td>Calculates the dot product of two 2D vectors</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_E</td>
<td>Represents the value of Euler's number</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_LineCircleIntersect(int in_x1, int in_y1, int in_x2, int in_y2, int in_circleX, int in_circleY, int in_radius)</td>
<td>Calculates the intersection of a line and a circle</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_LineRectangleIntersect(int in_x1, int in_y1, int in_x2, int in_y2, int in_rectX, int in_rectY, int in_rectW, int in_rectH)</td>
<td>Calculates the intersection of a line and a rectangle</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Multiply(int multiplicand, int multiplier)</td>
<td>Multiplies two integers</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Norm(int x, int y)</td>
<td>Calculates the norm of a 2D vector</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_NormPow(int x, int y)</td>
<td>Calculates the power of a 2D vector</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_PI</td>
<td>Represents the value of Pi</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_PointLineDistance(int in_lineX1, int in_lineY1, int in_lineX2, int in_lineY2, int in_pointX, int in_pointY)</td>
<td>Calculates the distance of a point from a line</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_RadiansToAngleFixedPoint(int angle)</td>
<td>Converts radians to angle fixed point</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_RadiansToDegrees(int angle)</td>
<td>Converts radians to degrees</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Round(int value)</td>
<td>Rounds an integer to the nearest integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Sqrt(int value)</td>
<td>Calculates the square root of a number</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Sqrt(long value)</td>
<td>Calculates the square root of a long integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Square(int value)</td>
<td>Calculates the square of a number</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPoint_Subtract(int minuend, int subtrahend)</td>
<td>Subtracts one integer from another integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPointAdjust(int a)</td>
<td>Adjusts an integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPointAngleToDegree(int a)</td>
<td>Converts an angle to degrees</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_FixedPointToInt(int a)</td>
<td>Converts an integer to a 32-bit integer</td>
<td>GLLib</td>
</tr>
<tr>
<td>Math_Init(String packName, int cos, int sqrt)</td>
<td>Initializes the GLLib package</td>
<td>GLLib</td>
</tr>
<tr>
<td>MATH_INTERSECT_NO_INTERSECT</td>
<td>Represents the absence of intersection</td>
<td>GLLib</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>MATH_INTERSECT_ONE_POINT</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>MATH_INTERSECT_TWO_POINTS</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_IntToFixedPoint(int a)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Log2(int a)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Max(int a, int b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Max(long a, long b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Min(int a, int b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Min(long a, long b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Norm(int x, int y, int iter)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_NormPow(int x1, int y1)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_QuickSort(int array[])</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_QuickSortIndices(int data[])</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_QuickSortIndices(int data[], int NbItemPerValue, int itemNb)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Quiet()</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Rand(int a, int b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Rand()</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_RandSetSeed(long seed)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_RectIntersect(int Ax0, int Ay0, int Ax1, int Ay1, int Bx0, int By0, int Bx1, int By1)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_SameSign(int a, int b)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_SegmentIntersect(int x1, int y1, int x2, int y2, int x3, int y3, int x4, int y4)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>MATH_SEGMENTINTERSECT_COLLINEAR</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>MATH_SEGMENTINTERSECT_DO_INTERSECT</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>MATH_SEGMENTINTERSECT_DONT_INTERSECT</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Sin(int a)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Sqrt(int x)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Sqrt(long val)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td><code>Math_Sqrt_FixedPoint(int val, int precisionLoop)</code></td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td>Package</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Math_Tan(int angle)</td>
<td>Calculate tangent of the angle</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_ArrayCopy(Object src, int src_position, Object dst, int dst_position, int length)</td>
<td>Copy array from src to dst</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_GetArray(byte[] src, int src_off, byte[] dst)</td>
<td>Get elements from src array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_GetByte(byte[] src, int src_off)</td>
<td>Get byte from src array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_GetInt(byte[] src, int src_off)</td>
<td>Get integer from src array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_GetLong(byte[] src, int src_off)</td>
<td>Get long from src array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_GetShort(byte[] src, int src_off)</td>
<td>Get short from src array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_ReadArray(InputStream is)</td>
<td>Read array from InputStream</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_SetArray(byte[] dst, int dst_off, byte[] src)</td>
<td>Set elements to dst array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_SetByte(byte[] dst, int dst_off, byte src)</td>
<td>Set byte to dst array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_SetInt(byte[] dst, int dst_off, int src)</td>
<td>Set integer to dst array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_SetLong(byte[] dst, int dst_off, long src)</td>
<td>Set long to dst array</td>
<td>GLLib</td>
</tr>
<tr>
<td>Mem_SetShort(byte[] dst, int dst_off, short src)</td>
<td>Set short to dst array</td>
<td>GLLib</td>
</tr>
<tr>
<td>MIME_type</td>
<td>Get MIME type</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Close()</td>
<td>Close pack file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_CloseShared()</td>
<td>Close shared pack file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_LoadMIME(String filename)</td>
<td>Load MIME file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Open(String filename)</td>
<td>Open pack file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_OpenShared(String strName, String strVendor, String strMidletName)</td>
<td>Open shared pack file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_PositionAtData(int idx)</td>
<td>Position at data</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Read()</td>
<td>Read pack file</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Read16()</td>
<td>Read 16-bit data</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Read32()</td>
<td>Read 32-bit data</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_ReadArray(int idx)</td>
<td>Read array at index</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_ReadData(int idx)</td>
<td>Read data at index</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_ReadFully(byte[] array, int offset, int length)</td>
<td>Read fully from array to offset length</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_ReleaseBinaryCache(String filename)</td>
<td>Release binary cache</td>
<td>GLLib</td>
</tr>
<tr>
<td>Pack_Seek(int addr)</td>
<td>Seek to position</td>
<td>GLLib</td>
</tr>
<tr>
<td>Function</td>
<td>Package</td>
<td>Modifier</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>Pack_Skip(int nb)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>paint(Graphics _g)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Pause()</td>
<td>GLLib</td>
<td>[protected]</td>
</tr>
<tr>
<td>PlatformRequest(String url)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>PosSlot</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Print(String log)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Profiler_BeginNamedEvent(String name)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Profiler_Draw()</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Profiler_End()</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Profiler_EndNamedEvent()</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>PROFILER_MAX_EVENTS</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Profiler_Start()</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Quit()</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>RepLenCoder</td>
<td>GLLib</td>
<td>[protected]</td>
</tr>
<tr>
<td>ResetAKey(int keyFlag)</td>
<td>GLLib</td>
<td>[static]</td>
</tr>
<tr>
<td>ResetKey()</td>
<td>GLLib</td>
<td>[static]</td>
</tr>
<tr>
<td>Resume()</td>
<td>GLLib</td>
<td>[protected]</td>
</tr>
<tr>
<td>RIGHT</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Rms_Read(String strName)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Rms_ReadShared(String strName, String strVendor, String strMidletName)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Rms_Write(String strName, byte[] data)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Rms_WriteShared(String strName, String strVendor, String strMidletName, byte[] data)</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>run()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>s_application</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_display</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_game_currentFrameNB</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_game_FPSAverage</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_game_frameDT</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_game_interruptNotify</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>s_game_isPaused</td>
<td>GLLib</td>
<td>[package, static]</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>s_game_keyEventIndex</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_keyJustPressed</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_keyPressedTime</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_lastFrameTime</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_state</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_timeWhenFrameStart</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_game_totalExecutionTime</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_gllib_instance</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_keyLastKeyPressUntranslatedCode</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_keyLastKeyStates</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_keyState</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_keyStateRT</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_bezierX</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_bezierY</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_bezierZ</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_Math_distPointLineX</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_Math_distPointLineY</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_F_05</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_F_1</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_Math_intersectPoints</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_Math_intersectX</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_Math_intersectY</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_math_random</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>s_pack_lastDataReadMimeType</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>SavePack(String packName, String rmsName)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>SetClip(int x, int y, int width, int height)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>setColor(int red, int green, int blue)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>SetColor(int RGB)</td>
<td>GLLib [package, static]</td>
<td></td>
</tr>
<tr>
<td>Method Name</td>
<td>Package</td>
<td>Modifier</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td><code>SetCurrentGraphics</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SetCurrentGraphics</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SetFont</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SetGrayScale</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SetStrokeStyle</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SetupDefaultKey</code></td>
<td>GLLib</td>
<td>protected</td>
</tr>
<tr>
<td><code>SetupDisplay</code></td>
<td>GLLib</td>
<td>protected</td>
</tr>
<tr>
<td><code>showNotify</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>sizeChanged</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SOLID</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>SpecPos</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Stream_Read</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Stream_Read16</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Stream_Read32</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Stream_ReadFully</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_BuildStringCache</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>text_encoding</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_FreeAll</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_FromUTF8</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_GetLanguageAsString</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_GetNbString</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_GetPhoneDefaultLangage</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_GetString</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_LoadTextFromPack</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>Text_LoadTextFromPack</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>tkNumMoveBits</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>TOP</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td><code>TRANS_MIRROR</code></td>
<td>GLLib</td>
<td>static</td>
</tr>
<tr>
<td>Method</td>
<td>Package</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT180</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT270</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_MIRROR_ROT90</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_NONE</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT180</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT270</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>TRANS_ROT90</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Translate(int x, int y)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>UnInit()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>VCENTER</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Vibrate(int duration)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>Warning(String message)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>WasAnyKeyPressed()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>WasAnyKeyReleased()</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>WasKeyPressed(int keyFlag)</td>
<td>GLLib</td>
<td></td>
</tr>
<tr>
<td>WasKeyReleased(int keyFlag)</td>
<td>GLLib</td>
<td></td>
</tr>
</tbody>
</table>
### GLLibConfig Member List

This is the complete list of members for **GLLibConfig**, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>disableNotifyDestroyed</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_Network_ErrorNoCredits</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_Network_ErrorNoNetwork</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_Network_ErrorNoNetworkAccess</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_Network_NoError</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_ScratchPad_CreditsFile</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_ScratchPad_EOF</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>doja_ScratchPad_SaveGameFile</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>FakeInterruptThreshold</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>FIXED_PRECISION</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>FPSLimiter</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeDown</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeFire</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeLeft</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeLeftSoftkey</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeRight</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeRightSoftkey</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>keycodeUp</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>lowMemoryLimit</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>math_angleFixedPointBase</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>math_AtanUseCacheTable</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>math_fixedPointBase</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>MAX_FLIP_COUNT</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>MAX_MODULE_MAPPINGS</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>MAX_SPRITE_PALETTESE</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>MAX_WRAP_TEXT_INFO</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>MIDP2forceNonFullScreen</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pack_dbgDataAccess</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pack_keepLoaded</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pack_skipbufferSize</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pack_supportLZMADecompression</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pack_useBlackBerryGZipDecompression</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pathfinding_Debug</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>pathfinding_MaxNode</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>platformRequestOnExit</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>PNG_BUFFER_SIZE</td>
<td>GLLibConfig [static]</td>
</tr>
<tr>
<td>Variable</td>
<td>File</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>rms_usePackRead</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>rms_useSharing</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>screenHeight</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>screenWidth</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>SLEEP_DRAWSTRINGB</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>softkeyOKOnLeft</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_enable</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_enableThread</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_numberOfChannels</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useCachedPlayers</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useFakeMediaDuration</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useFreeChannelOnStop</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useJSR135</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_usePrefetchedPlayers</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useRealizedPlayers</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useSetLevel</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useSetMediaTimeBeforePlay</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sound_useStopSoundsOnInterrupt</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_alwaysBsNfm1Byte</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_alwaysBsNoAfStart</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_alwaysBsNoFmStart</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_alwaysBsSkipFrameRc</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_animFPS</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_debugColision</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_debugErrors</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_debugLoading</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_debugModuleUsage</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_debugUsedMemory</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_drawPixelClippingBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_drawRegionFlippedBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_drawRGBTransparencyBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_fillRoundRectBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_fontBackslashChangePalette</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_fontUseOneFramePerLetter</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_fpsRegion</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_ModuleMapping_useModuleImages</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_newTextRendering</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_RGBArraysUseDrawRGB</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useAfOffShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useBSpriteFlags</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useBugFixImageOddSize</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useCacheFlipXY</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useCachePool</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>Function</td>
<td>Library</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>sprite_useCacheRGBArrays</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useCreateRGB</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useCreateRGBTransparencyBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useDeactivateSystemGc</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useDrawRegionClipping</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useDrawStringSleep</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useDynamicPng</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat127RLE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat16</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat2</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat256</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat256RLE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat4</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useEncodeFormat64RLE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useFMOffShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useFMPalette</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useFrameCollRC</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useFrameRects</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useGenPalette</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useGifHeader</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useHyperFM</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useIndexExAframes</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useIndexExFmodules</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useLoadImageWithoutTransf</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleColorAsByte</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleDataOffAsShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleMapping</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleUsageFromSprite</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleWHSShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleXY</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useModuleXYShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useMultipleModuleTypes</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useNokia7650DrawPixelBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useNokiaUI</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useNonInterlaced</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useOperationMark</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useOperationRecord</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useOperationRect</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_usePixelFormat0565</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_usePixelFormat1555</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_usePixelFormat4444</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Module Name</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>sprite_usePixelFormat8888</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_usePrecomputedCRC</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_usePrecomputedFrameRect</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useResize</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useSingleArrayForFMAF</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useSingleDirectGraphics</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useSingleImageForAllModules</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useSkipFastVisibilityTest</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useTransfFlip</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useTransfRot</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>sprite_useTruncatedRGBBuffer</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>text_useInternalUTF8Converter</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>text_useStringCache</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>tileset_maxLayerCount</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>tileset_useIndexAsShort</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>tileset_useTileShift</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>TMP_BUFFER_SIZE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useAbsoluteValueOfKeyCode</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useBugFixMultipleKeyPressed</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useCallSerially</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useDrawLineClippingBug</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useDrawPartialRGB</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useFakeInterruptHandling</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useFlashLightInsteadOfVibration</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useFrameDT</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useKeyAccumulation</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useSafeDrawRegion</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useSafeFillRect</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useServiceRepaints</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useSleepInsteadOfYield</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useSoftwareDoubleBuffer</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>useSystemGc</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_MXTELCEL</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_USCINGULAR_BLUE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_USCINGULAR_ORANGE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_USNEXTEL</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_USSPRINT</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CARRIER_USVIRGIN</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_CONN_TIMEOUT</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_DEBUG</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_DUAL_TCP</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_FIND_PLAYER</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_M7_SUPPORT</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>xplayer_ENABLE_MULTIPLAYER</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_PLAYER_SPECIFIC_DATA</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_ENABLE_TIMEOUT</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_HTTP_NO_CANCEL</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_KEEP_ALIVE_TIME</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_USE_BUG_FIX_MESSAGE_SIZE</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_USE_HTTP_POST</td>
<td>GLLibConfig</td>
</tr>
<tr>
<td>xplayer_XPLAYER_VERSION</td>
<td>GLLibConfig</td>
</tr>
</tbody>
</table>
GLLibMidiSpectrumAnalyzer Member List

This is the complete list of members for **GLLibMidiSpectrumAnalyzer**, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_INVALID_FILE_LENGTH</td>
<td>GLLibMidiSpectrumAnalyzer [static]</td>
</tr>
<tr>
<td>ERROR_INVALID_HEADER_CHUNK</td>
<td>GLLibMidiSpectrumAnalyzer [static]</td>
</tr>
<tr>
<td>ERROR_INVALID_META_EVENT</td>
<td>GLLibMidiSpectrumAnalyzer [static]</td>
</tr>
<tr>
<td>ERROR_INVALID_TRACK_CHUNK</td>
<td>GLLibMidiSpectrumAnalyzer [static]</td>
</tr>
<tr>
<td>GLLibMidiSpectrumAnalyzer(int bars, int maxValue, int velocityShift, int millisShift, int releaseDamping)</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>m_barLevels</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>m_bars</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>m_channelFilter</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>m_currentTime</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>MidiSpectrumAnalyzer_Parse(byte[] data)</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>MidiSpectrumAnalyzer_Reset()</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>MidiSpectrumAnalyzer_Update(long timeMillis)</td>
<td>GLLibMidiSpectrumAnalyzer</td>
</tr>
<tr>
<td>NO_ERRORS</td>
<td>GLLibMidiSpectrumAnalyzer [static]</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:32 2008 for GLib by doxygen 1.5.2
### GLLibPathFinding Member List

This is the complete list of members for `GLLibPathFinding`, including all inherited members.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GLLibPathFinding()</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>kDirDown</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirDownLeft</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirDownRight</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirLeft</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirRight</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirUp</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirUpLeft</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>kDirUpRight</code></td>
<td>GLLibPathFinding [static]</td>
</tr>
<tr>
<td><code>PathFinding_Execute(int start_x, int start_y, int start_dir, int end_x, int end_y)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_Free(boolean bKeepLastPath)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_GetPathLength()</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_GetPathPosition(int nIndex)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_GetPathPositionX(int nIndex)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_GetPathPositionY(int nIndex)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_Init(int nMapWidth, int nMapHeight, byte[] pPhysicalMap, int nCostMove, int nCostMoveDiag, int nCostChangeDir, int nDirCount)</code></td>
<td>GLLibPathFinding</td>
</tr>
<tr>
<td><code>PathFinding_Init(int nMapWidth, int nMapHeight, byte[] pPhysicalMap, int nCostMove, int nCostMoveDiag, int nCostChangeDir, int nDirCount, int nCollisionMask)</code></td>
<td>GLLibPathFinding</td>
</tr>
</tbody>
</table>
This is the complete list of members for **GLLibPlayer**, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>curFlags</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>curTime</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetAnim()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetDuration()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetFrame()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetNbanim()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetNbFrame()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetSprite()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GetTransform()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GLLibPlayer()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>GLLibPlayer(ASprite sprite, int x, int y)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>IsAnimOver()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>k_snd_priority_highest</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>k_snd_priority_lowest</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>k_snd_priority_normal</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>posX</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>posY</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>Render()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>Reset()</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>run()</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>s_data_mimeType</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>s_snd_masterVolume</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>s_snd_maxNbSoundSlot</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>SetAnim(int anim)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>SetAnim(int anim, int nbLoop)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>SetFrame(int frame)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>SetPos(int x, int y)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>SetSprite(ASprite sprite)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>SetTransform(int transform)</td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>Snd_DurationGet(int channel)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>Snd_ForceExecOnThreadOnGamePause()</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>Snd_FreeChannel(int channel)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Snd_GetChannelPlayer(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_GetChannelVolume(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_GetCurrentSoundIndex(int nChannel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_Init(int nbSoundSlot)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_IsPlaying(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_LoadSound(String dataFileName, int resourceIndex)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_LoadSound(String dataFileName, int resourceIndex, boolean bCacheThisSound)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_LoadSound(byte[] soundData, int nMIME, int index)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_LoadSound(byte[] soundData, int nMIME, int index, boolean bCacheThisSound)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_MediaTimeGet(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_MediaTimeSet(int channel, long time)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_MidiPlayNote(int channel, int MIDIChannel, int note, int volume)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_MidiSetChannelVolume(int channel, int MIDIChannel, int volume)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_Pause(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_Play(int channel, int index, int loop, int volume, int priority)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_PrepareSound(int channel, int index, int priority)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd.Quit()</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_RateGet(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_RateGetMax(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_RateGetMin(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_RateSet(int channel, int rate)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_Resume(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_SetMasterVolume(int volume)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_SetMediaDuration(int index, int duration)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_Stop(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_StopAllSounds()</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Snd_TempoGet(int channel)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>Function</td>
<td>Package</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td><code>Snd_TempoSet</code> (int channel, int tempo)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Snd_UnLoadSound</code> (int index)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Snd_Update</code> ()</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>sprite</code></td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td><code>Tileset_Destroy</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_Destroy</code> (int nLayer, boolean bFreeBufferImage)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_Draw</code> (Graphics g, int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_Draw</code> (Graphics g, int dx, int dy, int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetBufferGraphics</code> (int p_iLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetBufferGraphics</code> (int p_iLayer, int p_iImage)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetBufferImage</code> (int p_iLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetBufferImage</code> (int p_iLayer, int p_iImage)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetCamera</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetCameraX</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetCameraY</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetLayerHeight</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetLayerTileCountHeight</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetLayerTileCountWidth</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetLayerWidth</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetTile</code> (int nLayer, int x, int y)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_GetTileFlags</code> (int nLayer, int x, int y)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_Init</code> (int nDestWidth, int nDestHeight, int nTileWidth, int nTileHeight)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_LoadLayer</code> (int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, boolean bUseCB, int origin, int wrappingX, int wrappingY)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_LoadLayer</code> (int nLayer, byte[] MapSizes, byte[] MapData, byte[] MapFlip, ASprite MapSprite, int iUseCB, int origin, int wrappingX, int wrappingY)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_SetCamera</code> (int nLayer, int x, int y)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Tileset_Update</code> (int nLayer)</td>
<td>GLLibPlayer [package, static]</td>
</tr>
<tr>
<td><code>Update()</code></td>
<td>GLLibPlayer [package]</td>
</tr>
<tr>
<td>Function</td>
<td>Class</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Update(int DT)</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>WRAP_CLAMP</td>
<td>GLLibPlayer</td>
</tr>
<tr>
<td>WRAP_REPEAT</td>
<td>GLLibPlayer</td>
</tr>
</tbody>
</table>
# GRPH Member List

This is the complete list of members for **GRPH**, including all inherited members.

<table>
<thead>
<tr>
<th>Bottom</th>
<th>GRPH [static]</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCENTER</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>HCENTER_BOTTOM</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>HCENTER_TOP</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>HCENTER_VCENTER</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>Left</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>LEFT_BOTTOM</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>LEFT_TOP</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>LEFT_VCENTER</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>Right</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>RIGHT_BOTTOM</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>RIGHT_TOP</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>RIGHT_VCENTER</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>Top</td>
<td>GRPH [static]</td>
</tr>
<tr>
<td>VCenter</td>
<td>GRPH [static]</td>
</tr>
</tbody>
</table>

---

*Generated on Tue Sep 23 23:05:32 2008 for GLLib by doxygen 1.5.2*
HTTP Member List

This is the complete list of members for HTTP, including all inherited members.

<table>
<thead>
<tr>
<th>Function</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel()</td>
<td>HTTP</td>
</tr>
<tr>
<td>CarrierDeviceId</td>
<td>HTTP</td>
</tr>
<tr>
<td>cleanup()</td>
<td>HTTP</td>
</tr>
<tr>
<td>clientId</td>
<td>HTTP</td>
</tr>
<tr>
<td>isErrorOccurred()</td>
<td>HTTP</td>
</tr>
<tr>
<td>isInProgress()</td>
<td>HTTP</td>
</tr>
<tr>
<td>m_bError</td>
<td>HTTP</td>
</tr>
<tr>
<td>m_response</td>
<td>HTTP</td>
</tr>
<tr>
<td>m_responseByteArray</td>
<td>HTTP</td>
</tr>
<tr>
<td>run()</td>
<td>HTTP</td>
</tr>
<tr>
<td>sendByGet(String sUrl, String sQuery)</td>
<td>HTTP</td>
</tr>
<tr>
<td>upsubid</td>
<td>HTTP</td>
</tr>
<tr>
<td>userAgent</td>
<td>HTTP</td>
</tr>
<tr>
<td>x_up_calling_line_id</td>
<td>HTTP</td>
</tr>
<tr>
<td>x_up_subno</td>
<td>HTTP</td>
</tr>
</tbody>
</table>
License Member List

This is the complete list of members for **License**, including all inherited members.

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>callstarttime</td>
<td>License  [static]</td>
</tr>
<tr>
<td>cancel()</td>
<td>License</td>
</tr>
<tr>
<td>cleanup()</td>
<td>License</td>
</tr>
<tr>
<td>getLastError()</td>
<td>License</td>
</tr>
<tr>
<td>handleValidateLicense()</td>
<td>License</td>
</tr>
<tr>
<td>isLicenseValid()</td>
<td>License</td>
</tr>
<tr>
<td>License(MIDlet midlet)</td>
<td>License</td>
</tr>
<tr>
<td>NOT_A_NUMBER</td>
<td>License  [static]</td>
</tr>
<tr>
<td>sendValidateLicense()</td>
<td>License</td>
</tr>
</tbody>
</table>
License.Error Member List

This is the complete list of members for License.Error, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_BAD_RESPONSE</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_CONNECTION</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_INIT</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_INVALID_GGI</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_CLIENT_ID</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_PHONE_NUMBER</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_UUID</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_NONE</td>
<td>License.Error [static]</td>
</tr>
<tr>
<td>ERROR_PENDING</td>
<td>License.Error [static]</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:32 2008 for GLib by doxygen 1.5.2
**MessageType Member List**

This is the complete list of members for *MessageType*, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC_MESSAGE_TYPE_CONNECT</td>
<td>MessageType</td>
</tr>
<tr>
<td>BASIC_MESSAGE_TYPE_GAME</td>
<td>MessageType</td>
</tr>
<tr>
<td>BASIC_MESSAGE_TYPE_UNKNOWN</td>
<td>MessageType</td>
</tr>
<tr>
<td>CONNECT_MESSAGE_DISCONNECT</td>
<td>MessageType</td>
</tr>
<tr>
<td>CONNECT_MESSAGE_ERROR</td>
<td>MessageType</td>
</tr>
<tr>
<td>CONNECT_MESSAGE_INIT_READ</td>
<td>MessageType</td>
</tr>
<tr>
<td>CONNECT_MESSAGE_INIT_WRITE</td>
<td>MessageType</td>
</tr>
<tr>
<td>CONNECT_MESSAGE_SUCCESS</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_BLOCKED_SESSION</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_CREATE_SESSION_INVALID_NAME</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_CREATE_SESSION_NAME_USED</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_INVALID_INPUT</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_KICK_OUT_PLAYER_IS_MASTER</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_LIST_SESSION_INVALID_INDEX</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_LOGIN_AUTHENTICATION_FAILED</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_LOGIN_INVALID_NICKNAME</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_LOGIN_NICKNAME_USED</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_NOPLAYER</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_NO_SESSION</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_NOT_MASTER</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_SESSION_CLOSED</td>
<td>MessageType</td>
</tr>
<tr>
<td>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_ERROR</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_IN_GAME</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_IN_GAME_TOALL</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_KEEP_ALIVE</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_PUSH</td>
<td>MessageType</td>
</tr>
<tr>
<td>GAME_MESSAGE_REQUEST</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_HAS_NO_PSD</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_HAS_PSD</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_NOT_REGISTERED</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_OFFLINE</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_ONLINE</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_ONLINE_IN_SESSION</td>
<td>MessageType</td>
</tr>
<tr>
<td>PLAYER_STATUS_ONLINE_PLAYING</td>
<td>MessageType</td>
</tr>
<tr>
<td>Message Type</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_FINISH_GAME</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_JOIN_SESSION</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_KICK_OUT</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_LEAVE_SESSION</td>
<td></td>
</tr>
<tr>
<td>PUSH_MESSAGE_START_GAME</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_CREATE_SESSION</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_FINISH_GAME</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_DATA</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_INFO</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_QUICK_SESSION</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_JOIN_SESSION</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_KICK_OUT_PLAYER</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LEAVE_SESSION</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LIST_SESSIONS</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LOGIN</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_PROXY</td>
<td></td>
</tr>
<tr>
<td>REQUEST_MESSAGE_START_GAME</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_ERROR</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_SUCCESS</td>
<td></td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:33 2008 for GLLib by doxygen 1.5.2
<table>
<thead>
<tr>
<th>Main Page</th>
<th>Modules</th>
<th>Packages</th>
<th>Classes</th>
<th>Files</th>
<th>Related Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alphabetical List</td>
<td>Class List</td>
<td>Class Hierarchy</td>
<td>Class Members</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TCP Member List

This is the complete list of members for TCP, including all inherited members.

<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>connect()</td>
<td>TCP</td>
</tr>
<tr>
<td>connection</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>disconnect()</td>
<td>TCP</td>
</tr>
<tr>
<td>incomingQueue</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>istream</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>m_bError</td>
<td>TCP</td>
</tr>
<tr>
<td>m_bErrorString</td>
<td>TCP</td>
</tr>
<tr>
<td>m_connected</td>
<td>TCP</td>
</tr>
<tr>
<td>ostream</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>outgoing_data_on_receive_connection</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>outgoingQueue</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>receive_connection</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>receive_istream</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>receive_ostream</td>
<td>TCP [protected]</td>
</tr>
<tr>
<td>recvPacket()</td>
<td>TCP</td>
</tr>
<tr>
<td>run()</td>
<td>TCP</td>
</tr>
<tr>
<td>sendEstablishConnectionPackageOnReceive(byte[] data)</td>
<td>TCP</td>
</tr>
<tr>
<td>sendPacket(byte[] data)</td>
<td>TCP</td>
</tr>
<tr>
<td>TCP(String url)</td>
<td>TCP [protected]</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:33 2008 for GLib by doxygen 1.5.2
XPlayer Member List

This is the complete list of members for **XPlayer**, including all inherited members.

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPlayer.addByte(byte data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addByteArray(byte size, int dataOffset, byte[] data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addInt(int data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addMultipleScoreEntry(int score, int level, int scoreType)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addMultipleScoreEntryWithSupplementalData(int score, int level, int scoreType, int[] supplemental_data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addShort(short data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.addString(String data)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.callstarttime</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.cancel()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.cleanup()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.clearData()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getByte()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getByteArray()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getByteArray(int dataOffset, byte[] datadest)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getDataList()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getCurrentPlayerLeaderboardPosition()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getCurrentPlayerLeaderboardScore()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getCurrentPlayerLeaderboardScoreData()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getData()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getDataList()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getFirstSessionIndex()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getFoundPlayerName()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getFoundPlayerSessionName()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getFoundPlayerSessionNumberOfPlayers()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getFoundPlayerStatus()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getInt()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLastError()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardData(String[] names, int[] positions, int[] scores, int[][] scoreDatas)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardEntryPlayerName(int index)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardEntryPlayerPosition(int index)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardEntryPlayerScore(int index)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardEntryPlayerScoreData(int index)</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLeaderboardSize()</td>
<td>XPlayer member</td>
</tr>
<tr>
<td>XPlayer.getLength()</td>
<td>XPlayer member</td>
</tr>
</tbody>
</table>

*Note: Some methods are marked with `[static]` indicating they are static methods.*
<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMyAvgScore()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyBestRank()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyHighScore()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyHighScoreData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyLastTimePlayed()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyLowScore()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyLowScoreData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getMyNumberOfGamesPlayed()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getNameList()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getNewRankAfterScoreSending()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getNumberOfItems()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getPlayerStats(int[] bestRankArr, int[] highScoreArr, int[] lowScoreArr, int[] avgScoreArr, int[] numberOfGamesPlayedArr, String[] lastTimePlayedArr, int[] lowScoreDataArr, int[] highScoreDataArr)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getRequestedPlayerData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getRequestedPlayerNickname()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getSessionData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getSessionName()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getShort()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getString()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>getUsername()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleChangeUsername()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleHighscore()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleLogin()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleRankGet()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleRankGetAroundPlayer()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleRateGame()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleRecommendGame()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>handleStatsGet()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>initMultipleScores()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>isGameMessageInQueue</td>
<td>XPlayer</td>
</tr>
<tr>
<td>isLoggedIn()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpDisconnect()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleCreateSession()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleDisconnect()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleEstablishConnection()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleFindPlayer()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleFinishGame()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleGameData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleGetPlayerData()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleJoinSession()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleKickOutPlayer(String name)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleLeaveSession()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>mpHandleListSession()</td>
<td>XPlayer</td>
</tr>
<tr>
<td>mpHandleLogin()</td>
<td></td>
</tr>
<tr>
<td>mpHandleStartGame()</td>
<td></td>
</tr>
<tr>
<td>mpHandleUpdates()</td>
<td></td>
</tr>
<tr>
<td>mpHasOpponentFinished()</td>
<td></td>
</tr>
<tr>
<td>mplsConnected()</td>
<td></td>
</tr>
<tr>
<td>mplsInGame()</td>
<td></td>
</tr>
<tr>
<td>mplsInSession()</td>
<td></td>
</tr>
<tr>
<td>mplsLoggedIn()</td>
<td></td>
</tr>
<tr>
<td>mplsIsMaster()</td>
<td></td>
</tr>
<tr>
<td>mpPrepareGameData()</td>
<td></td>
</tr>
<tr>
<td>mpSendCreateSession(String sessionname, String sessiondata)</td>
<td></td>
</tr>
<tr>
<td>mpSendDisconnect()</td>
<td></td>
</tr>
<tr>
<td>mpSendEstablishConnection()</td>
<td></td>
</tr>
<tr>
<td>mpSendFindPlayer(String playerName)</td>
<td></td>
</tr>
<tr>
<td>mpSendFinishGame()</td>
<td></td>
</tr>
<tr>
<td>mpSendGameData()</td>
<td></td>
</tr>
<tr>
<td>mpSendGetPlayerData(String playerName)</td>
<td></td>
</tr>
<tr>
<td>mpSendJoinSession(String sessionname)</td>
<td></td>
</tr>
<tr>
<td>mpSendKickOutPlayer(String playerName)</td>
<td></td>
</tr>
<tr>
<td>mpSendLeaveSession()</td>
<td></td>
</tr>
<tr>
<td>mpSendListSession(byte numberOfSessions, byte firstSessionIndex)</td>
<td></td>
</tr>
<tr>
<td>mpSendLogin()</td>
<td></td>
</tr>
<tr>
<td>mpSendQuickGame()</td>
<td></td>
</tr>
<tr>
<td>mpSendStartGame()</td>
<td></td>
</tr>
<tr>
<td>NOT_A_NUMBER</td>
<td>XPlayer     [static]</td>
</tr>
<tr>
<td>SCORE_TYPE_POINTS</td>
<td>XPlayer     [static]</td>
</tr>
<tr>
<td>SCORE_TYPE_TIME</td>
<td>XPlayer     [static]</td>
</tr>
<tr>
<td>sendChangeUsername()</td>
<td></td>
</tr>
<tr>
<td>sendHighscore(int score, int level, int scoreType)</td>
<td></td>
</tr>
<tr>
<td>sendHighscoreWithSupplementalData(int score, int level, int scoreType, int[] supplemental_data)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>sendLogin(String playerData)</td>
<td></td>
</tr>
<tr>
<td>sendMultipleHighscores()</td>
<td></td>
</tr>
<tr>
<td>sendRankGet(int level, int number_of_players, int scoreType, int numberOfSupplementalDataFields)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>sendRankGetAroundPlayer(int level, int number_of_players_around, int scoreType, int numberOfSupplementalDataFields)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>sendRateGame(int rating)</td>
<td></td>
</tr>
<tr>
<td>sendRecommendGame()</td>
<td></td>
</tr>
<tr>
<td>sendStatsGet(int level, int numberOfSupplementalDataFields)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>setData(byte[] data)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Class</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>setPhoneNumber(String phoneNr)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>setUsername(String name)</td>
<td>XPlayer</td>
</tr>
<tr>
<td>XPlayer(MIDlet midlet)</td>
<td>XPlayer</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:33 2008 for GLib by doxygen 1.5.2
# XPlayer.ConnectLevel Member List

This is the complete list of members for `XPlayer.ConnectLevel`, including all inherited members.

<table>
<thead>
<tr>
<th>Member</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN_CONNECTED</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>IN_GAME</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>IN_GAME_MASTER</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>IN_SESSION</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>IN_SESSION_MASTER</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>NOT_CONNECTED</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
<tr>
<td>NOT_LOGGED_IN</td>
<td><code>XPlayer.ConnectLevel</code> [static]</td>
</tr>
</tbody>
</table>

Generated on Tue Sep 23 23:05:33 2008 for GLib by doxygen 1.5.2
XPlayer.Error Member List

This is the complete list of members for XPlayer.Error, including all inherited members.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_BAD_RESPONSE</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_CHANGE_USERNAME_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_CONNECTION</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_GET_RANKINGS_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_GET_STATS_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_INIT</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_INVALID_GGI</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_JOIN_GAME</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NICK_TAKEN</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_CLIENT_ID</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_NICKNAME</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_PHONE_NUMBER</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NO_UUID</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NONE</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NOT_M7_ENABLED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_NOT_REGISTERED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_PENDING</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_RATE_GAME_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_RECOMMEND_GAME_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_REGISTER_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_REQUEST_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_SCORE_UPLOAD_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_START_GAME</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_SUPPLEMENTAL_DATA_NEEDED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_VALIDATE_LICENSE_FAILED</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_WRONGFULL_QSTATE</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>ERROR_WRONGFULL_RSTATE</td>
<td>XPlayer.Error [static]</td>
</tr>
<tr>
<td>Main Page</td>
<td>Modules</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>Alphabetical List</td>
<td>Class List</td>
</tr>
</tbody>
</table>
# XPlayer.MessageType Member List

This is the complete list of members for `XPlayer.MessageType`, including all inherited members.

<table>
<thead>
<tr>
<th>Basic Message Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BASIC_MESSAGE_TYPE_CONNECT</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>BASIC_MESSAGE_TYPE_GAME</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>BASIC_MESSAGE_TYPE_UNKNOWN</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connection Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CONNECT_MESSAGE_DISCONNECT</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>CONNECT_MESSAGE_INIT_READ</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>CONNECT_MESSAGE_INIT_WRITE</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>CONNECT_MESSAGE_SUCCESS</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ERROR_BLOCKED_SESSION</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_CREATE_SESSION_INVALID_NAME</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_CREATE_SESSION_NAME_USED</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_INVALID_INPUT</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_JOIN_SESSION_TOO_MANY_PLAYERS</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_KICK_OUT_PLAYER_IS_MASTER</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_LIST_SESSION_INVALID_INDEX</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_LOGIN_AUTHENTICATION_FAILED</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_LOGIN_INVALID_NICKNAME</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_LOGIN_NICKNAME_USED</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_NOPLAYER</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_NO_SESSION</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_NOT_MASTER</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_SESSION_CLOSED</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>ERROR_START_GAME_NOT_ENOUGH_PLAYERS</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Game Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>GAME_MESSAGE_ERROR</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>GAME_MESSAGE_IN_GAME</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>GAME_MESSAGE_IN_GAME_TOALL</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>GAME_MESSAGE_KEEP_ALIVE</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>GAME_MESSAGE_PUSH</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>GAME_MESSAGE_REQUEST</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Player Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PLAYER_STATUS_HAS_NO_PSD</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_HAS_PSD</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_NOT_REGISTERED</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_OFFLINE</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_ONLINE</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_ONLINE_IN_SESSION</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td><code>PLAYER_STATUS_ONLINE_PLAYING</code></td>
<td>XPlayer.MessageType [static]</td>
</tr>
<tr>
<td>Message Type</td>
<td>Class</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>PUSH_MESSAGE_FINISH_GAME</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>PUSH_MESSAGE_JOIN_SESSION</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>PUSH_MESSAGE_KICK_OUT</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>PUSH_MESSAGE_LEAVE_SESSION</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>PUSH_MESSAGE_START_GAME</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_CREATE_SESSION</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_DATA</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_PLAYER_INFO</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_GET_QUICK_SESSION</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_KICK_OUT_PLAYER</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LEAVE_SESSION</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LIST_SESSIONS</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_LOGIN</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>REQUEST_MESSAGE_START_GAME</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_ERROR</td>
<td>XPlayer.MessageType</td>
</tr>
<tr>
<td>RESPONSE_MESSAGE_SUCCESS</td>
<td>XPlayer.MessageType</td>
</tr>
</tbody>
</table>
- [-

- [static initializer](): \texttt{ASprite}, \texttt{GLLib}
- _aframes : ASprite
  - _aframes_flags : ASprite
  - _aframes_frame : ASprite
  - _aframes_ox_byte : ASprite
  - _aframes_ox_short : ASprite
  - _aframes_oy_byte : ASprite
  - _aframes_oy_short : ASprite
  - _aframes_time : ASprite
  - _alpha : ASprite
  - _anims_af_start : ASprite
  - _anims_naf : ASprite
  - _aryPrecomputedFlags : ASprite
  - _aryPrecomputedImages : ASprite
  - _aryPrecomputedImgX : ASprite
  - _aryPrecomputedImgY : ASprite
  - _aryPrecomputedSizeX : ASprite
  - _aryPrecomputedSizeY : ASprite
  - _aryPrecomputedX : ASprite
  - _aryPrecomputedY : ASprite
  - _bs_flags : ASprite
  - _bTraceNow : ASprite
  - _buffer_index : ASprite
  - _colors : ASprite
  - _cur_pal : ASprite
  - _cur_pool : ASprite
  - _data_format : ASprite
  - _fmodules : ASprite
  - _fmodules_flags : ASprite
  - _fmodules_id : ASprite
  - _fmodules_ox_byte : ASprite
- _fmodules_ox_short : ASprite
- _fmodules_oy_byte : ASprite
- _fmodules_oy_short : ASprite
- _fmodules_pal : ASprite
- _frames_col : ASprite
- _frames_col_short : ASprite
- _frames_fm_start : ASprite
- _frames_nfm : ASprite
- _frames_rc : ASprite
- _frames_rc_short : ASprite
- _frames_rects : ASprite
- _frames_rects_short : ASprite
- _frames_rects_start : ASprite
- _gifHeader : ASprite
- _graphics : ASprite
- _header_size : ASprite
- _i64rle_color_bits : ASprite
- _i64rle_color_mask : ASprite
- _images_count : ASprite
- _images_size : ASprite
- _index1 : ASprite
- _index2 : ASprite
- _itoa_buffer : ASprite
- _main_image : ASprite
- _map : ASprite
- _module_colors_byte : ASprite
- _module_colors_int : ASprite
- _module_image_imageAA : ASprite
- _module_image_imageAAA : ASprite
- _module_image_intAAA : ASprite
- _module_types : ASprite
- _modules_data : ASprite
- _modules_data_off_int : ASprite
- _modules_data_off_short : ASprite
- _modules_extra_info : ASprite
- _modules_extra_pointer : ASprite
- _modules_h_byte : ASprite
- _modules_h_scaled : ASprite
- _modules_h_short : ASprite
• _modules_image_shortAAA : ASprite
• _modules_usage : ASprite
• _modules_w_byte : ASprite
• _modules_w_scaled : ASprite
• _modules_w_short : ASprite
• _modules_x_byte : ASprite
• _modules_x_short : ASprite
• _modules_y_byte : ASprite
• _modules_y_short : ASprite
• _nModules : ASprite
• _old_pal : ASprite
• _operation : ASprite
• _pal_data : ASprite
• _pal_int : ASprite
• _pal_short : ASprite
• _palettes : ASprite
• _png_index : ASprite
• _PNG_packed_IDAT_ADLER : ASprite
• _PNG_packed_IDAT_CRC : ASprite
• _PNG_packed_IHDR_CRC : ASprite
• _PNG_packed_PLTE_CRC : ASprite
• _PNG_packed_tRNS_CRC : ASprite
• _png_result : ASprite
• _png_size : ASprite
• _png_start_crc : ASprite
• _poolCacheSprites : ASprite
• _poolCacheStack : ASprite
• _poolCacheStackIndex : ASprite
• _poolCacheStackMax : ASprite
• _rectX1 : ASprite
• _rectX2 : ASprite
• _rectY1 : ASprite
• _rectY2 : ASprite
• _text_h : ASprite
• _text_w : ASprite
• _transp : ASprite
• _w_pos : ASprite
• _warpTextInfo : ASprite
Here is a list of all class members with links to the classes they belong to:

- _aframes : ASprite
- _aframes_flags : ASprite
- _aframes_frame : ASprite
- _aframes_ox_byte : ASprite
- _aframes_ox_short : ASprite
- _aframes_oy_byte : ASprite
- _aframes_oy_short : ASprite
- _aframes_time : ASprite
- _alpha : ASprite
- _anims_af_start : ASprite
- _anims_naf : ASprite
- _aryPrecomputedFlags : ASprite
- _aryPrecomputedImages : ASprite
- _aryPrecomputedImgX : ASprite
- _aryPrecomputedImgY : ASprite
- _aryPrecomputedSizeX : ASprite
- _aryPrecomputedSizeY : ASprite
- _aryPrecomputedX : ASprite
- _aryPrecomputedY : ASprite
- _bs_flags : ASprite
- _bTraceNow : ASprite
- _buffer_index : ASprite
- _colors : ASprite
- _cur_pal : ASprite
- _cur_pool : ASprite
- _data_format : ASprite
- _fmodules : ASprite
- _fmodules_flags : ASprite
- _fmodules_id : ASprite
- `_fmodules_ox_byte : ASprite`
- `_fmodules_ox_short : ASprite`
- `_fmodules_oy_byte : ASprite`
- `_fmodules_oy_short : ASprite`
- `_fmodules_pal : ASprite`
- `_frames_col : ASprite`
- `_frames_col_short : ASprite`
- `_frames_fm_start : ASprite`
- `_frames_nfm : ASprite`
- `_frames_rc : ASprite`
- `_frames_rc_short : ASprite`
- `_frames_rects : ASprite`
- `_frames_rects_short : ASprite`
- `_frames_rects_start : ASprite`
- `_gifHeader : ASprite`
- `_graphics : ASprite`
- `_header_size : ASprite`
- `_i64rle_color_bits : ASprite`
- `_i64rle_color_mask : ASprite`
- `_images_count : ASprite`
- `_images_size : ASprite`
- `_index1 : ASprite`
- `_index2 : ASprite`
- `_itoa_buffer : ASprite`
- `_main_image : ASprite`
- `_map : ASprite`
- `_module_colors_byte : ASprite`
- `_module_colors_int : ASprite`
- `_module_image_imageAA : ASprite`
- `_module_image_imageAAA : ASprite`
- `_module_image_intAAA : ASprite`
- `_module_types : ASprite`
- `_modules_data : ASprite`
- `_modules_data_off_int : ASprite`
- `_modules_data_off_short : ASprite`
- `_modules_extra_info : ASprite`
- `_modules_extra_pointer : ASprite`
- `_modules_h_byte : ASprite`
- `_modules_h_scaled : ASprite`
- _modules_h_short: ASprite
- _modules_image_shortAAA: ASprite
- _modules_usage: ASprite
- _modules_w_byte: ASprite
- _modules_w_scaled: ASprite
- _modules_w_short: ASprite
- _modules_x_byte: ASprite
- _modules_x_short: ASprite
- _modules_y_byte: ASprite
- _modules_y_short: ASprite
- _nModules: ASprite
- _old_pal: ASprite
- _operation: ASprite
- _pal_data: ASprite
- _pal_int: ASprite
- _pal_short: ASprite
- _palettes: ASprite
- _png_index: ASprite
- _PNG_packed_IDAT_ADLER: ASprite
- _PNG_packed_IDAT_CRC: ASprite
- _PNG_packed_IHDR_CRC: ASprite
- _PNG_packed_PLTE_CRC: ASprite
- _PNG_packed_tRNS_CRC: ASprite
- _png_result: ASprite
- _png_size: ASprite
- _png_start_crc: ASprite
- _poolCacheSprites: ASprite
- _poolCacheStack: ASprite
- _poolCacheStackIndex: ASprite
- _poolCacheStackMax: ASprite
- _rectX1: ASprite
- _rectX2: ASprite
- _rectY1: ASprite
- _rectY2: ASprite
- _text_h: ASprite
- _text_w: ASprite
- _transp: ASprite
- _w_pos: ASprite
- _warpTextInfo: ASprite
Here is a list of all class members with links to the classes they belong to:

- **a** -

  - addByte() : [XPlayer](#)
  - addByteArray() : [XPlayer](#)
  - addInt() : [XPlayer](#)
  - addMultipleScoreEntry() : [XPlayer](#)
  - addMultipleScoreEntryWithSupplementalData() : [XPlayer](#)
  - addShort() : [XPlayer](#)
  - addString() : [XPlayer](#)
  - Align : [GLLib](#)
  - ASprite() : [ASprite](#)
  - Assert() : [GLLib](#)
Here is a list of all class members with links to the classes they belong to:

- **b** -

- **BASE : ASprite**
- **BASIC_MESSAGE_TYPE_CONNECT :** `MessageType` , `XPlayer.MessageType`
- **BASIC_MESSAGE_TYPE_GAME :** `MessageType` , `XPlayer.MessageType`
- **BASIC_MESSAGE_TYPE_UNKNOWN :** `XPlayer.MessageType` , `MessageType`
- **BLOCK_INFO_SIZE : ASprite**
- **BOTTOM : GLLib , GRPH**
- **BR : GLLang**
- **BS_AF_OFF_SHORT : ASprite**
- **BS_ANIMS : ASprite**
- **BS_DEFAULT_DOJA : ASprite**
- **BS_DEFAULT_MIDP1 : ASprite**
- **BS_DEFAULT_MIDP1b : ASprite**
- **BS_DEFAULT_MIDP1c : ASprite**
- **BS_DEFAULT_MIDP2 : ASprite**
- **BS_DEFAULT_NOKIA : ASprite**
- **BS_FM_OFF_SHORT : ASprite**
- **BS_FM_PALETTE : ASprite**
- **BS_FRAME_COLL_RC : ASprite**
- **BS_FRAME_RECTS : ASprite**
- **BS_FRAMES : ASprite**
- **BS_GIF_HEADER : ASprite**
- **BS_IMAGE_SIZE_INT : ASprite**
- **BS_KEEP_PAL : ASprite**
- **BS_MODULE_IMAGES : ASprite**
- **BS_MODULE_IMAGES_FX : ASprite**
- **BS_MODULE_USAGE : ASprite**
- BS_MODULES : ASprite
- BS_MODULES_IMG : ASprite
- BS_MODULES_USAGE : ASprite
- BS_MODULES_WH_SHORT : ASprite
- BS_MODULES_XY : ASprite
- BS_MODULES_XY_SHORT : ASprite
- BS_NAF_1_BYTE : ASprite
- BS_NFM_1_BYTE : ASprite
- BS_NO_AF_START : ASprite
- BS_PNG_CRC : ASprite
- BS_SINGLE_IMAGE : ASprite
- BS_SKIP_FRAME_RC : ASprite
- BS_TRANSP_FIRST : ASprite
- BS_TRANSP_LAST : ASprite
- BSPRITE_v003 : ASprite
- BSPRITE_v004 : ASprite
- BSPRITE_v005 : ASprite
- BuildAnimCacheImages() : ASprite
- BuildCacheImages() : ASprite
- BuildFrameCacheImages() : ASprite
Here is a list of all class members with links to the classes they belong to:

- **c** -

- callstarttime: [License, XPlayer]
- cancel(): [XPlayer, HTTP, License]
- CarrierDeviceId: [HTTP]
- CheckAndDumpConfig(): [GLLib]
- cleanup(): [License, XPlayer, HTTP]
- clearData(): [XPlayer]
- clientId: [HTTP]
- ClipRect(): [GLLib]
- CN: [GLLang]
- connect(): [TCP]
- CONNECT_MESSAGE_DISCONNECT: [XPlayer.MessageType, MessageType]
- CONNECT_MESSAGE_ERROR: [XPlayer.MessageType, MessageType]
- CONNECT_MESSAGE_INIT_READ: [XPlayer.MessageType, MessageType]
- CONNECT_MESSAGE_INIT_WRITE: [XPlayer.MessageType, MessageType]
- CONNECT_MESSAGE_SUCCESS: [XPlayer.MessageType, MessageType]
- connection: [TCP]
- ConvertFixedPointToString(): [GLLib]
- CopyArea(): [GLLib]
- CountFrameModules(): [ASprite]
- Crc32(): [ASprite]
- CRC32_POLYNOMIAL: [ASprite]
- crcTable: [ASprite]
- curFlags: [GLLibPlayer]
- currentChunkType: [ASprite]
Here is a list of all class members with links to the classes they belong to:

- d -

- **Dbg()**: GLLib
- **DE**: GLlang
- **DecodeImage()**: ASprite
- **DecodeImageToByteArray()**: ASprite
- **disableNotifyDestroyed**: GLLibConfig
- **disconnect()**: TCP
- **doja_Network_ErrorNoCredits**: GLLibConfig
- **doja_Network_ErrorNoNetwork**: GLLibConfig
- **doja_Network_ErrorNoNetworkAccess**: GLLibConfig
- **doja_Network_NoError**: GLLibConfig
- **doja_ScratchPad_CreditsFile**: GLLibConfig
- **doja_ScratchPad_EOF**: GLLibConfig
- **doja_ScratchPad_SaveGameFile**: GLLibConfig
- **DOTTED**: GLLib
- **DrawArc()**: GLLib
- **DrawChar()**: GLLib
- **DrawChars()**: GLLib
- **DrawImage()**: GLLib
- **DrawLine()**: ASprite, GLLib
- **DrawNumber()**: ASprite
- **DrawPage()**: ASprite
- **DrawPageB()**: ASprite
- **drawPartialRGB()**: GLLib
- **DrawRect()**: GLLib
- **DrawRegion()**: GLLib
- **DrawRGB()**: GLLib
- **DrawRoundRect()**: GLLib
- **DrawString()**: ASprite, GLLib, ASprite
- **DrawStringOrChars()**: ASprite
DrawSubstring() : GLib

Generated on Tue Sep 23 23:05:31 2008 for GLib by doxygen 1.5.2
Here is a list of all class members with links to the classes they belong to:

- e -

- **EN**: [GLLang](#)
- ENCODE_FORMAT_I127RLE: ASprite
- ENCODE_FORMAT_I16: ASprite
- ENCODE_FORMAT_I2: ASprite
- ENCODE_FORMAT_I256: ASprite
- ENCODE_FORMAT_I256RLE: ASprite
- ENCODE_FORMAT_I4: ASprite
- ENCODE_FORMAT_I64RLE: ASprite
- ERROR_BAD_RESPONSE: License.Error, XPlayer.Error
- ERROR_BLOCKED_SESSION: MessageType, XPlayer.MessageType
- ERROR_CHANGE_USERNAME_FAILED: XPlayer.Error
- ERROR_CONNECTION: XPlayer.Error, License.Error
- ERROR_CREATE_SESSION_INVALID_NAME: XPlayer.MessageType, MessageType
- ERROR_CREATE_SESSION_NAME_USED: XPlayer.MessageType, MessageType
- ERROR_GET_RANKINGS_FAILED: XPlayer.Error
- ERROR_GET_STATS_FAILED: XPlayer.Error
- ERROR_INIT: XPlayer.Error, License.Error
- ERROR_INVALID_FILE_LENGTH: GLLibMidiSpectrumAnalyzer
- ERROR_INVALID_GGI: License.Error, XPlayer.Error
- ERROR_INVALID_HEADER_CHUNK: GLLibMidiSpectrumAnalyzer
- ERROR_INVALID_INPUT: XPlayer.MessageType, MessageType
- ERROR_INVALID_META_EVENT: GLLibMidiSpectrumAnalyzer
- ERROR_INVALID_TRACK_CHUNK: GLLibMidiSpectrumAnalyzer
- ERROR_JOIN_GAME: XPlayer.Error
- ERROR_JOIN_SESSION_TOO_MANY_PLAYERS: MessageType, XPlayer.MessageType
- ERROR_KICK_OUT_PLAYER_IS_MASTER: MessageType, XPlayer.MessageType
- ERROR_LIST_SESSION_INVALID_INDEX: MessageType, XPlayer.MessageType
- ERROR_LOGIN_AUTHENTICATION_FAILED: XPlayer.MessageType, MessageType
- ERROR_LOGIN_INVALID_NICKNAME: MessageType, XPlayer.MessageType
- ERROR_LOGIN_NICKNAME_USED: XPlayer.MessageType
- ERROR_NICK_TAKEN: XPlayer.Error
- ERROR_NO_CLIENT_ID: License.Error, XPlayer.Error
- ERROR_NO_NICKNAME: XPlayer.Error
- ERROR_NO_PHONE_NUMBER: License.Error, XPlayer.Error
- ERROR_NO_PLAYER: MessageType, XPlayer.MessageType
- ERROR_NO_SESSION: XPlayer.MessageType, MessageType
- ERROR_NO_UUID: XPlayer.Error, License.Error
- ERROR_NONE: XPlayer.Error, License.Error
- ERROR_NOT_M7_ENABLED: XPlayer.Error
- ERROR_NOT_MASTER: XPlayer.MessageType, MessageType
- ERROR_NOT_REGISTERED: XPlayer.Error
- ERROR_PENDING: License.Error, XPlayer.Error
- ERROR_RATE_GAME_FAILED: XPlayer.Error
- ERROR_RECOMMEND_GAME_FAILED: XPlayer.Error
- ERROR_REGISTER_FAILED: XPlayer.Error
- ERROR_REQUEST_FAILED: XPlayer.Error
- ERROR_SCORE_UPLOAD_FAILED: XPlayer.Error
- ERROR_SESSION_CLOSED: MessageType, XPlayer.MessageType
- ERROR_START_GAME: XPlayer.Error
- ERROR_START_GAME_NOT_ENOUGH_PLAYERS: XPlayer.MessageType, MessageType
- ERROR_SUPPLEMENTAL_DATA_NEEDED: XPlayer.Error
- ERROR_VALIDATE_LICENSE_FAILED: XPlayer.Error
- ERROR_WRONGFULL_QSTATE: XPlayer.Error
- ERROR_WRONGFULL_RSTATE: XPlayer.Error
- ES: GLLang
Here is a list of all class members with links to the classes they belong to:

- **f** -

- FakeInterruptThreshold : [GLLibConfig](#)
- FillArc() : [GLLib](#)
- FillRect() : [GLLib](#)
- FillRoundRect() : [GLLib](#)
- FillTriangle() : [GLLib](#)
- FIXED_PRECISION : [GLLibConfig](#)
- FLAG_FLIP_X : [ASprite](#)
- FLAG_FLIP_Y : [ASprite](#)
- FLAG_HYPER_FM : [ASprite](#)
- FLAG_INDEX_EX_MASK : [ASprite](#)
- FLAG_OFFSET_AF : [ASprite](#)
- FLAG_OFFSET_FM : [ASprite](#)
- FLAG_ROT_90 : [ASprite](#)
- FLAG_USER0 : [ASprite](#)
- FLAG_USER1 : [ASprite](#)
- FPSLimiter : [GLLibConfig](#)
- FR : [GLLang](#)
- FreeAnimCacheImages() : [ASprite](#)
- FreeCacheData() : [ASprite](#)
- FreeFrameCacheImages() : [ASprite](#)
- FreeMemory() : [ASprite](#)
- FreeModuleImage() : [ASprite](#)
Here is a list of all class members with links to the classes they belong to:

- **g** -
  
  - `g : GLLib`  
  - `Game_KeyClearKeyCode() : GLLib`  
  - `Game_KeySetKeyCode() : GLLib`  
  - `GAME_MESSAGE_ERROR : MessageType, XPlayer.MessageType`  
  - `GAME_MESSAGE_IN_GAME : MessageType, XPlayer.MessageType`  
  - `GAME_MESSAGE_IN_GAME_TOALL : MessageType, XPlayer.MessageType`  
  - `GAME_MESSAGE_KEEP_ALIVE : XPlayer.MessageType, MessageType`  
  - `GAME_MESSAGE_PUSH : XPlayer.MessageType, MessageType`  
  - `GAME_MESSAGE_REQUEST : XPlayer.MessageType, MessageType`  
  - `Game_Run() : GLLib`  
  - `Game_update() : GLLib`  
  - `Gc() : GLLib`  
  - `GenPalette() : ASprite`  
  - `GetAFrameFlags() : ASprite`  
  - `GetAFrameRect() : ASprite`  
  - `GetFrames() : ASprite`  
  - `GetFramesOX() : ASprite`  
  - `GetFramesOY() : ASprite`  
  - `GetFrameTime() : ASprite`  
  - `GetAnim() : GLLibPlayer`  
  - `GetAnimFrame() : ASprite`  
  - `GetBlueComponent() : GLLib`  
  - `GetBold() : ASprite`  
  - `getByte() : XPlayer`
- `getByteArray() : XPlayer`
- `getByteDataList() : XPlayer`
- `GetCharFrame() : ASprite`
- `GetCharMap() : ASprite`
- `GetCharMapShort() : ASprite`
- `GetChars() : ASprite`
- `getCharSize() : ASprite`
- `GetCharSpacing() : ASprite`
- `GetClipHeight() : GLLib`
- `GetClipWidth() : GLLib`
- `GetClipX() : GLLib`
- `GetClipY() : GLLib`
- `GetColor() : GLLib`
- `GetCurrentMMapping() : ASprite`
- `GetCurrentPalette() : ASprite`
- `getCurrentPlayerLeaderboardPosition() : XPlayer`
- `getCurrentPlayerLeaderboardScore() : XPlayer`
- `getCurrentPlayerLeaderboardScoreData() : XPlayer`
- `GetCurrentStringHeight() : ASprite`
- `GetCurrentStringWidth() : ASprite`
- `getData() : XPlayer`
- `getDataList() : XPlayer`
- `GetDisplayColor() : GLLib`
- `GetDuration() : GLLibPlayer`
- `getFirstSessionIndex() : XPlayer`
- `GetFModuleOX() : ASprite`
- `GetFModuleOY() : ASprite`
- `GetFModuleRect() : ASprite`
- `GetFModules() : ASprite`
- `GetFont() : GLLib`
- `GetFontHeight() : ASprite`
- `getFoundPlayerName() : XPlayer`
- `getFoundPlayerSessionName() : XPlayer`
- `getFoundPlayerSessionNumberOfPlayers() : XPlayer`
- `getFoundPlayerStatus() : XPlayer`
- `GetFrame() : GLLibPlayer`
- `GetFrameCount() : ASprite`
- `GetFrameHeight() : ASprite`
- `GetFrameMarkers() : ASprite`
• GetFrameModule() : ASprite
• GetFrameModuleFlags() : ASprite
• GetFrameModuleHeight() : ASprite
• GetFrameModulePalette() : ASprite
• GetFrameModuleWidth() : ASprite
• GetFrameModuleX() : ASprite
• GetFrameModuleY() : ASprite
• GetFrameRect() : ASprite
• GetFrameRectCount() : ASprite
• GetFrames() : ASprite
• GetFrameTime() : GLLib
• GetFrameWidth() : ASprite
• GetGrayScale() : GLLib
• GetGreenComponent() : GLLib
• getInt() : XPlayer
• getLastError() : License, XPlayer
• getLeaderboardData() : XPlayer
• getLeaderboardEntryPlayerName() : XPlayer
• getLeaderboardEntryPlayerPosition() : XPlayer
• getLeaderboardEntryPlayerScore() : XPlayer
• getLeaderboardEntryPlayerScoreData() : XPlayer
• getLeaderboardSize() : XPlayer
• getLength() : XPlayer
• GetLineHeight() : ASprite
• GetLineSpacing() : ASprite
• GetMIME() : GLLib
• GetModuleCount() : ASprite
• GetModuleData() : ASprite
• GetModuleHeight() : ASprite
• GetModuleHeightOrg() : ASprite
• GetModuleImage() : ASprite
• GetModuleRect() : ASprite
• GetModuleWidth() : ASprite
• GetModuleWidthOrg() : ASprite
• GetModuleX() : ASprite
• GetModuleY() : ASprite
• getMyAvgScore() : XPlayer
• getMyBestRank() : XPlayer
• getMyHighScore() : XPlayer
- getMyHighScoreData(): XPlayer
- getMyLastTimePlayed(): XPlayer
- getMyLowScore(): XPlayer
- getMyLowScoreData(): XPlayer
- getMyNumberOfGamesPlayed(): XPlayer
- getNameList(): XPlayer
- GetNbanim(): GLLibPlayer
- GetNBData(): GLLib
- GetNbFrame(): GLLibPlayer
- getNewRankAfterScoreSending(): XPlayer
- getNumberOfItems(): XPlayer
- GetPalette(): ASprite
- getPlayerStats(): XPlayer
- GetRealTime(): GLLib
- GetRedComponent(): GLLib
- getRequestedPlayerData(): XPlayer
- getRequestedPlayerNickname(): XPlayer
- GetResourceAsStream(): GLLib
- GetScreenHeight(): GLLib
- GetScreenWidth(): GLLib
- getSessionData(): XPlayer
- getSessionName(): XPlayer
- getShort(): XPlayer
- GetSoftwareDoubleBuffer(): GLLib
- GetSoftwareDoubleBufferGraphics(): GLLib
- GetSpaceWidth(): ASprite
- GetSprite(): GLLibPlayer
- getString(): XPlayer
- GetStrokeStyle(): GLLib
- GetTransform(): GLLibPlayer
- GetTranslateX(): GLLib
- GetTranslateY(): GLLib
- GetUnderline(): ASprite
- getUsername(): XPlayer
- GLLib(): GLLib
- GLLibMidiSpectrumAnalyzer(): GLLibMidiSpectrumAnalyzer
- GLLibPathFinding(): GLLibPathFinding
- GLLibPlayer(): GLLibPlayer
Here is a list of all class members with links to the classes they belong to:

- h -

- handleChangeUsername() : XPlayer
- handleHighscore() : XPlayer
- handleLogin() : XPlayer
- handleRankGet() : XPlayer
- handleRankGetAroundPlayer() : XPlayer
- handleRateGame() : XPlayer
- handleRecommendGame() : XPlayer
- handleStatsGet() : XPlayer
- handleValidateLicense() : License
- HCENTER : GRPH, GLLib
- HCENTER_BOTTOM : GRPH
- HCENTER_TOP : GRPH
- HCENTER_VCENTER : GRPH
- HEADER_LEVEL0_MAX_WBITS : ASprite
- hideNotify() : GLLib
- hRef : ASprite
- hTarget : ASprite
Here is a list of all class members with links to the classes they belong to:

- i -

- IDAT : ASprite
- IEND : ASprite
- IHDR : ASprite
- IN_CONNECTED : XPlayer.ConnectLevel
- IN_GAME : XPlayer.ConnectLevel
- IN_GAME_MASTER : XPlayer.ConnectLevel
- IN_SESSION : XPlayer.ConnectLevel
- IN_SESSION_MASTER : XPlayer.ConnectLevel
- incomingQueue : TCP
- INDEX_EX_MASK : ASprite
- INDEX_EX_SHIFT : ASprite
- INDEX_MASK : ASprite
- INFO32 : ASprite
- INFO8 : ASprite
- Init() : GLLib
- InitCachePool() : ASprite
- initMultipleScores() : XPlayer
- InitPoolSize() : ASprite
- InitSharedRms() : GLLib
- inputIndex : GLLib
- IsAnimOver() : GLLibPlayer
- IsAnyKeyDown() : GLLib
- isErrorOccurred() : HTTP
- isGameMessageInQueue : XPlayer
- isInProgress() : HTTP
- IsKeyDown() : GLLib
- IsKeyUp() : GLLib
- isLicenseValid() : License
- isLoggedln() : XPlayer
- IsMatch : **GLLib**
- IsRep : **GLLib**
- IsRep0Long : **GLLib**
- IsRepG0 : **GLLib**
- IsRepG1 : **GLLib**
- IsRepG2 : **GLLib**
- istream : **TCP**
- IT : **GLLang**

---

*Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2*
Here is a list of all class members with links to the classes they belong to:

- j -

- **JP**: GLLang

*Generated on Tue Sep 23 23:05:31 2008 for GLib by [doxygen](http://www.stackoverflow.com) 1.5.2*
Here is a list of all class members with links to the classes they belong to:

- **k** -

  - k_consoleLogEnable : GLKey
  - k_down : GLKey
  - k_dummy : GLKey
  - k_fire : GLKey
  - k_invalid : GLKey
  - k_itoa_buffer_size : ASprite
  - k_left : GLKey
  - k_menuBack : GLKey
  - k_menuOK : GLKey
  - k_nbKey : GLKey
  - k_num0 : GLKey
  - k_num1 : GLKey
  - k_num2 : GLKey
  - k_num3 : GLKey
  - k_num4 : GLKey
  - k_num5 : GLKey
  - k_num6 : GLKey
  - k_num7 : GLKey
  - k_num8 : GLKey
  - k_num9 : GLKey
  - k_pound : GLKey
  - k_right : GLKey
  - k_rmsLogEnable : GLKey
  - k_snd_priority_highest : GLLibPlayer
  - k_snd_priority_lowest : GLLibPlayer
  - k_snd_priority_normal : GLLibPlayer
  - k_star : GLKey
  - k_up : GLKey
  - kAlignTableSize : GLLib
- kBitModelTotal : GLLib
- kDirDown : GLLibPathFinding
- kDirDownLeft : GLLibPathFinding
- kDirDownRight : GLLibPathFinding
- kDirLeft : GLLibPathFinding
- kDirRight : GLLibPathFinding
- kDirUp : GLLibPathFinding
- kDirUpLeft : GLLibPathFinding
- kDirUpRight : GLLibPathFinding
- kEndPosModelIndex : GLLib
- keycodeDown : GLLibConfig
- keycodeFire : GLLibConfig
- keycodeLeft : GLLibConfig
- keycodeLeftSoftkey : GLLibConfig
- keycodeRight : GLLibConfig
- keycodeRightSoftkey : GLLibConfig
- keycodeUp : GLLibConfig
- keyPressed() : GLLib
- keyReleased() : GLLib
- kLenNumHighBits : GLLib
- kLenNumHighSymbols : GLLib
- kLenNumLowBits : GLLib
- kLenNumLowSymbols : GLLib
- kLenNumMidBits : GLLib
- kLenNumMidSymbols : GLLib
- kMatchMinLen : GLLib
- kNumAlignBits : GLLib
- kNumBitModelTotalBits : GLLib
- kNumFullDistances : GLLib
- kNumLenProbs : GLLib
- kNumLenToPosStates : GLLib
- kNumMoveBits : GLLib
- kNumPosBitsMax : GLLib
- kNumPosSlotBits : GLLib
- kNumPosStatesMax : GLLib
- kNumStates : GLLib
- kNumTopBits : GLLib
- KR : GLLang
- kStartPosModelIndex : GLLib
• kTopValue : **GLLib**
Here is a list of all class members with links to the classes they belong to:

- I -

- LEFT : `GLLib`, `GRPH`
- LEFT_BOTTOM : `GRPH`
- LEFT_TOP : `GRPH`
- LEFT_VCENTER : `GRPH`
- LenChoice : `GLLib`
- LenChoice2 : `GLLib`
- LenCoder : `GLLib`
- LenHigh : `GLLib`
- LenLow : `GLLib`
- LenMid : `GLLib`
- License() : `License`
- Literal : `GLLib`
- Load() : `ASprite`
- lowMemoryLimit : `GLLibConfig`
- LZMA_BASE_SIZE : `GLLib`
- LZMA_Inflate() : `GLLib`
- LZMA_LIT_SIZE : `GLLib`
- LZMA_RESULT_DATA_ERROR : `GLLib`
- LZMA_RESULT_NOT_ENOUGH_MEM : `GLLib`
- LZMA_RESULT_OK : `GLLib`
Here is a list of all class members with links to the classes they belong to:

- m -

- m_barLevels : GLLibMidiSpectrumAnalyzer
- m_bars : GLLibMidiSpectrumAnalyzer
- m_bError : HTTP, TCP
- m_bErrorString : TCP
- m_Buffer : GLLib
- m_channelFilter : GLLibMidiSpectrumAnalyzer
- m_Code : GLLib
- m_connected : TCP
- m_current_keys_pressed : GLLib
- m_current_keys_released : GLLib
- m_current_keys_state : GLLib
- m_currentTime : GLLibMidiSpectrumAnalyzer
- m_ExtraBytes : GLLib
- m_inSize : GLLib
- m_keys_pressed : GLLib
- m_keys_released : GLLib
- m_keys_state : GLLib
- m_last_key_pressed : GLLib
- m_outStream : GLLib
- m_Range : GLLib
- m_response : HTTP
- m_responseByteArray : HTTP
- MAGIC : ASprite
- MAGIC_IDAT_h : ASprite
- MAGIC_IEND : ASprite
- Math_Abs() : GLLib
- Math_Angle180 : GLLib
- Math_Angle270 : GLLib
- Math_Angle360 : GLLib
- Math_Angle90 : GLLib
- math_angleFixedPointBase : GLLibConfig
- Math_AngleMUL : GLLib
- Math_Atan() : GLLib
- Math_AtanSlow() : GLLib
- math_AtanUseCacheTable : GLLibConfig
- Math_Bezier2D() : GLLib
- Math_Bezier3D() : GLLib
- Math_Cos() : GLLib
- Math_DegreeToFixedPointAngle() : GLLib
- Math_Det() : GLLib
- Math_DistPointLine() : GLLib
- Math_Div10() : GLLib
- Math_DotProduct() : GLLib
- Math_FixedPoint_Add() : GLLib
- Math_FixedPoint_DegreesToAngleFixedPoint() : GLLib
- Math_FixedPoint_DegreesToRadians() : GLLib
- Math_FixedPoint_Det() : GLLib
- Math_FixedPoint_Divide() : GLLib
- Math_FixedPoint_DotProduct() : GLLib
- Math_FixedPoint_E : GLLib
- Math_FixedPoint_LineCircleIntersect() : GLLib
- Math_FixedPoint_LineRectangleIntersect() : GLLib
- Math_FixedPoint_Multiply() : GLLib
- Math_FixedPoint_Norm() : GLLib
- Math_FixedPoint_NormPow() : GLLib
- Math_FixedPoint_PI : GLLib
- Math_FixedPoint_PointLineDistance() : GLLib
- Math_FixedPoint_RadiansToAngleFixedPoint() : GLLib
- Math_FixedPoint_RadiansToDegrees() : GLLib
- Math_FixedPoint_Round() : GLLib
- Math_FixedPoint_Sqrt() : GLLib
- Math_FixedPoint_Square() : GLLib
- Math_FixedPoint_Subtract() : GLLib
- Math_FixedPointAdjust() : GLLib
- Math_FixedPointAngleToDegree() : GLLib
- math_fixedPointBase : GLLibConfig
- Math_FixedPointToInt() : GLLib
- Math_Init() : GLLib
- MATH_INTERSECT_NO_INTERSECT : **GLLib**
- MATH_INTERSECT_ONE_POINT : **GLLib**
- MATH_INTERSECT_TWO_POINTS : **GLLib**
- Math_IntToFixedPoint() : **GLLib**
- Math_Log2() : **GLLib**
- Math_Max() : **GLLib**
- Math_Min() : **GLLib**
- Math_Norm() : **GLLib**
- Math_NormPow() : **GLLib**
- Math_QuickSort() : **GLLib**
- Math_QuickSortIndices() : **GLLib**
- Math_Quit() : **GLLib**
- Math_Rand() : **GLLib**
- Math_RandSetSeed() : **GLLib**
- Math_RectIntersect() : **GLLib**
- Math_SameSign() : **GLLib**
- Math_SegmentIntersect() : **GLLib**
- MATH_SEGMENTINTERSECT_COLLINEAR : **GLLib**
- MATH_SEGMENTINTERSECT_DO_INTERSECT : **GLLib**
- MATH_SEGMENTINTERSECT_DONT_INTERSECT : **GLLib**
- Math_Sin() : **GLLib**
- Math_Sqrt() : **GLLib**
- Math_Sqrt_FixedPoint() : **GLLib**
- Math_Tan() : **GLLib**
- MAX_FLIP_COUNT : **GLLibConfig**
- MAX_MODULE_MAPPINGS : **GLLibConfig**
- MAX_SPRITE_PALETTE : **GLLibConfig**
- MAX_TRANSFORMATION_FLAGS : **ASprite**
- MAX_WRAP_TEXT_INFO : **GLLibConfig**
- MD_ARC : **ASprite**
- MD_FILL_ARC : **ASprite**
- MD_FILL_RECT : **ASprite**
- MD_FILL_TRIANGLE : **ASprite**
- MD_IMAGE : **ASprite**
- MD_MARKER : **ASprite**
- MD_RECT : **ASprite**
- MD_TRIANGLE : **ASprite**
- mem : **ASprite**
- Mem_ArrayCopy() : **GLLib**
- Mem_GetArray() : GLLib
- Mem_GetByte() : GLLib
- Mem_GetInt() : GLLib
- Mem_GetLong() : GLLib
- Mem_GetShort() : GLLib
- Mem_ReadArray() : GLLib
- Mem_SetArray() : GLLib
- Mem_SetByte() : GLLib
- Mem_SetInt() : GLLib
- Mem_SetLong() : GLLib
- Mem_SetShort() : GLLib
- MidiSpectrumAnalyzer_Parse() : GLLibMidiSpectrumAnalyzer
- MidiSpectrumAnalyzer_Reset() : GLLibMidiSpectrumAnalyzer
- MidiSpectrumAnalyzer_Update() : GLLibMidiSpectrumAnalyzer
- midp2_flags : ASprite
- MIDP2forceNonFullScreen : GLLibConfig
- MIME_type : GLLib
- mod : ASprite
- ModifyPalette() : ASprite
- ModifyPaletteAlpha() : ASprite
- ModifyPaletteAlphaUsingAltPalette() : ASprite
- ModifyPaletteAlphaUsingLastPalette() : ASprite
- mpDisconnect() : XPlayer
- mpHandleCreateSession() : XPlayer
- mpHandleDisconnect() : XPlayer
- mpHandleEstablishConnection() : XPlayer
- mpHandleFindPlayer() : XPlayer
- mpHandleFinishGame() : XPlayer
- mpHandleGameData() : XPlayer
- mpHandleGetPlayerData() : XPlayer
- mpHandleJoinSession() : XPlayer
- mpHandleKickOutPlayer() : XPlayer
- mpHandleLeaveSession() : XPlayer
- mpHandleListSession() : XPlayer
- mpHandleLogin() : XPlayer
- mpHandleStartGame() : XPlayer
- mpHandleUpdates() : XPlayer
- mpHasOpponnentFinished() : XPlayer
- mpIsConnected() : XPlayer
- mpIsInGame() : XPlayer
- mpIsInSession() : XPlayer
- mpIsLoggedIn() : XPlayer
- mpIsMaster() : XPlayer
- mpPrepareGameData() : XPlayer
- mpSendCreateSession() : XPlayer
- mpSendDisconnect() : XPlayer
- mpSendEstablishConnection() : XPlayer
- mpSendFindPlayer() : XPlayer
- mpSendFinishGame() : XPlayer
- mpSendGameData() : XPlayer
- mpSendGetPlayerData() : XPlayer
- mpSendJoinSession() : XPlayer
- mpSendKickOutPlayer() : XPlayer
- mpSendLeaveSession() : XPlayer
- mpSendListSession() : XPlayer
- mpSendLogin() : XPlayer
- mpSendQuickGame() : XPlayer
- mpSendStartGame() : XPlayer
- mResizeCorrectY : ASprite
- mResizeRef : ASprite
Here is a list of all class members with links to the classes they belong to:

- n -

- NMAX : ASprite
- NO_ERRORS : GLLibMidiSpectrumAnalyzer
- NOT_A_NUMBER : XPlayer, License
- NOT_CONNECTED : XPlayer.ConnectLevel
- NOT_LOGGED_IN : XPlayer.ConnectLevel
Here is a list of all class members with links to the classes they belong to:

- o -

- OPERATION_COMPUTERECT : ASprite
- OPERATION_DRAW : ASprite
- OPERATION_MARK : ASprite
- OPERATION_RECORD : ASprite
- ostream : TCP
- outgoing_data_on_receive_connection : TCP
- outgoingQueue : TCP

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
Here is a list of all class members with links to the classes they belong to:

- p -

- Pack_Close() : GLLib
- Pack_CloseShared() : GLLib
- pack_dbgDataAccess : GLLibConfig
- pack_keepLoaded : GLLibConfig
- Pack_LoadMIME() : GLLib
- Pack_Open() : GLLib
- Pack_OpenShared() : GLLib
- Pack_PositionAtData() : GLLib
- Pack_Read() : GLLib
- Pack_Read16() : GLLib
- Pack_Read32() : GLLib
- Pack_ReadArray() : GLLib
- Pack_ReadData() : GLLib
- Pack_ReadFully() : GLLib
- Pack_ReleaseBinaryCache() : GLLib
- Pack_SeeK() : GLLib
- Pack_Skip() : GLLib
- pack_skipbufferSize : GLLibConfig
- pack_supportLZMADecompression : GLLibConfig
- pack_useBlackBerryGZipDecompression : GLLibConfig
- paint() : GLLib
- PaintAFrame() : ASprite
- PaintFModule() : ASprite
- PaintFrame() : ASprite
- PaintModule() : ASprite
- PaintPrecomputedFrame() : ASprite
- PAL_BLEND_BLACK : ASprite
- PAL_BLUE_CYAN : ASprite
- PAL_GREEN : ASprite
- PAL_GREY : ASprite
- PAL_INVISIBLE : ASprite
- PAL_ORIGINAL : ASprite
- PAL_RED_YELLOW : ASprite
- pathfinding_Debug : GLLibConfig
- PathFinding_Exec() : GLLibPathFinding
- PathFinding_Free() : GLLibPathFinding
- PathFinding_GetPathLength() : GLLibPathFinding
- PathFinding_GetPathPosition() : GLLibPathFinding
- PathFinding_GetPathPositionX() : GLLibPathFinding
- PathFinding_GetPathPositionY() : GLLibPathFinding
- PathFinding_Init() : GLLibPathFinding
- pathfinding_MaxNode : GLLibConfig
- Pause() : GLLib
- PIXEL_FORMAT_0565 : ASprite
- PIXEL_FORMAT_1555 : ASprite
- PIXEL_FORMAT_4444 : ASprite
- PIXEL_FORMAT_8888 : ASprite
- PL : GLLang
- PlatformRequest() : GLLib
- platformRequestOnExit : GLLibConfig
- PLAYER_STATUS_HAS_NO_PSD : MessageType, XPlayer.MessageType
- PLAYER_STATUS_HAS_PSD : MessageType, XPlayer.MessageType
- PLAYER_STATUS_NOT_REGISTERED : XPlayer.MessageType, MessageType
- PLAYER_STATUS_OFFLINE : XPlayer.MessageType, MessageType
- PLAYER_STATUS_ONLINE : MessageType, XPlayer.MessageType
- PLAYER_STATUS_ONLINE_IN_SESSION : MessageType, XPlayer.MessageType
- PLAYER_STATUS_ONLINE_PLAYING : MessageType, XPlayer.MessageType
- PLTE : ASprite
- PNG_BUFFER_SIZE : GLLibConfig
- PNG_INFO_SIZE : ASprite
- PosSlot : GLLib
- posX : **GLLibPlayer**
- posY : **GLLibPlayer**
- PrecomputeAllFrames() : **ASprite**
- PrecomputeFrame() : **ASprite**
- Print() : **GLLib**
- Profiler_BeginNamedEvent() : **GLLib**
- Profiler_Draw() : **GLLib**
- Profiler_End() : **GLLib**
- Profiler_EndNamedEvent() : **GLLib**
- PROFILER_MAX_EVENTS : **GLLib**
- Profiler_Start() : **GLLib**
- PT : **GLLang**
- PUSH_MESSAGE_FINISH_GAME : **XPlayer.MessageType**
- PUSH_MESSAGE_JOIN_SESSION : **XPlayer.MessageType**
- PUSH_MESSAGE_KICK_OUT : **XPlayer.MessageType**
- PUSH_MESSAGE_LEAVE_SESSION : **XPlayer.MessageType**
- PUSH_MESSAGE_START_GAME : **XPlayer.MessageType**
Here is a list of all class members with links to the classes they belong to:

- q -

  - Quit() : GLLib
Here is a list of all class members with links to the classes they belong to:

- r -

- receive_connection : TCP
- receive_istream : TCP
- receive_ostream : TCP
- record_frame : ASprite
- record_index : ASprite
- recvPacket() : TCP
- Render() : GLLibPlayer
- RepLenCoder : GLLib
- REQUEST_MESSAGE_CREATE_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_FINISH_GAME : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_PLAYER_DATA : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_PLAYER_INFO : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_QUICK_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_JOIN_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_KICK_OUT_PLAYER : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LEAVE_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LIST_SESSIONS : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LOGIN : MessageType , XPlayer.MessageType
- REQUEST_MESSAGE_PROXY : MessageType
- REQUEST_MESSAGE_START_GAME: XPlayer.MessageType, MessageType
- Reset(): GLLibPlayer
- ResetAKey(): GLLib
- ResetCachePool(): ASprite
- ResetKey(): GLLib
- RESIZE_CREATERGB: ASprite
- RESIZE_DRAW_ON_MUTABLE: ASprite
- RESIZE_NONE: ASprite
- RESIZE_NOT_CACHED: ASprite
- RESIZE_REF_176x220: ASprite
- RESIZE_REF_240x256: ASprite
- RESIZE_REF_240x320: ASprite
- ResizeCoords(): ASprite
- RESPONSE_MESSAGE_ERROR: MessageType, XPlayer.MessageType
- RESPONSE_MESSAGE_SUCCESS: MessageType, XPlayer.MessageType
- Resume(): GLLib
- RIGHT: GLLib, GRPH
- RIGHT_BOTTOM: GRPH
- RIGHT_TOP: GRPH
- RIGHT_VCENTER: GRPH
- Rms_Read(): GLLib
- Rms_ReadShared(): GLLib
- rms_usePackRead: GLLibConfig
- rms_useSharing: GLLibConfig
- Rms_Write(): GLLib
- Rms_WriteShared(): GLLib
- RU: GLLang
- run(): HTTP, TCP, GLLibPlayer, GLLib
Here is a list of all class members with links to the classes they belong to:

- **s** -

  - `s_application` : **GLLib**
  - `s_bAspectRatio` : **ASprite**
  - `s_bBilinear` : **ASprite**
  - `s_data_mimeType` : **GLLibPlayer**
  - `s_display` : **GLLib**
  - `s_game_currentFrameNB` : **GLLib**
  - `s_game_FPSAverage` : **GLLib**
  - `s_game_frameDT` : **GLLib**
  - `s_game_interruptNotify` : **GLLib**
  - `s_game_isPaused` : **GLLib**
  - `s_game_keyEventIndex` : **GLLib**
  - `s_game_keyJustPressed` : **GLLib**
  - `s_game_keyPressedTime` : **GLLib**
  - `s_game_lastFrameTime` : **GLLib**
  - `s_game_state` : **GLLib**
  - `s_game_timeWhenFrameStart` : **GLLib**
  - `s_game_totalExecutionTime` : **GLLib**
  - `s_gllib_instance` : **GLLib**
  - `s_keyLastKeyPressUntranslatedCode` : **GLLib**
  - `s_keyLastKeyStates` : **GLLib**
  - `s_keyState` : **GLLib**
  - `s_keyStateRT` : **GLLib**
  - `s_MapChar` : **ASprite**
  - `s_math_bezierX` : **GLLib**
  - `s_math_bezierY` : **GLLib**
  - `s_math_bezierZ` : **GLLib**
  - `s_Math_distPointLineX` : **GLLib**
  - `s_Math_distPointLineY` : **GLLib**
  - `s_math_F_05` : **GLLib**
- s_math_F_1 : GLLib
- s_Math_intersectPoints : GLLib
- s_Math_intersectX : GLLib
- s_Math_intersectY : GLLib
- s_math_random : GLLib
- s_pack_lastDataReadMimeType : GLLib
- s_rc : ASprite
- s_resizeType : ASprite
- s_snd_masterVolume : GLLibPlayer
- s_snd_maxNbSoundSlot : GLLibPlayer
- SavePack() : GLLib
- SCALE_SHIFT : ASprite
- scaleX() : ASprite
- scaleY() : ASprite
- SCORE_TYPE_POINTS : XPlayer
- SCORE_TYPE_TIME : XPlayer
- screenHeight : GLLibConfig
- screenWidth : GLLibConfig
- sendByGet() : HTTP
- sendChangeUsername() : XPlayer
- sendEstablishConnectionPackageOnReceive() : TCP
- sendHighscore() : XPlayer
- sendHighscoreWithSupplementalData() : XPlayer
- sendLogin() : XPlayer
- sendMultipleHighscores() : XPlayer
- sendPacket() : TCP
- sendRankGet() : XPlayer
- sendRankGetAroundPlayer() : XPlayer
- sendRateGame() : XPlayer
- sendRecommendGame() : XPlayer
- sendStatsGet() : XPlayer
- sendValidateLicense() : License
- SetAnim() : GLLibPlayer
- SetBold() : ASprite
- SetCharMap() : ASprite
- SetCharMapStatic() : ASprite
- SetCharSpacing() : ASprite
- SetCharSpacingToDefault() : ASprite
- SetClip() : GLLib
- SetColor() : GLLib
- setColor() : GLLib
- SetCurrentGraphics() : GLLib
- SetCurrentMMapping() : ASprite
- SetCurrentPalette() : ASprite
- setData() : XPlayer
- SetDefaultFontMetrics() : ASprite
- SetFont() : GLLib
- setFrame() : GLLibPlayer
- SetGraphics() : ASprite
- SetGrayScale() : GLLib
- SetLineHeight() : ASprite
- SetLineHeightToDefault() : ASprite
- SetLineSpacing() : ASprite
- SetLineSpacingToDefault() : ASprite
- SetModuleImage() : ASprite
- SetModuleImagesArray() : ASprite
- SetModuleMapping() : ASprite
- setPhoneNumber() : XPlayer
- SetPool() : ASprite
- SetPos() : GLLibPlayer
- SetResizeParameters() : ASprite
- SetSpaceWidth() : ASprite
- SetSpaceWidthToDefault() : ASprite
- Sprite() : GLLibPlayer
- SetStrokeStyle() : GLLib
- SetSubString() : ASprite
- SetTempBuffer() : ASprite
- SetTransform() : GLLibPlayer
- SetUnderline() : ASprite
- SetupDefaultKey() : GLLib
- SetupDisplay() : GLLib
- setUsername() : XPlayer
- showNotify() : GLLib
- sizeChanged() : GLLib
- SLEEP_DRAWSTRINGB : GLLibConfig
- Snd_DurationGet() : GLLibPlayer
- Snd_ForceExecOnThreadOnGamePause() : GLLibPlayer
- Snd_FreeChannel() : GLLibPlayer
- `Snd_GetChannelPlayer()`: `GLLibPlayer`
- `Snd_GetChannelVolume()`: `GLLibPlayer`
- `Snd_GetCurrentSoundIndex()`: `GLLibPlayer`
- `Snd_Init()`: `GLLibPlayer`
- `Snd_IsPlaying()`: `GLLibPlayer`
- `Snd_LoadSound()`: `GLLibPlayer`
- `Snd_MediaTimeGet()`: `GLLibPlayer`
- `Snd_MediaTimeSet()`: `GLLibPlayer`
- `Snd_MidiPlayNote()`: `GLLibPlayer`
- `Snd_MidiSetChannelVolume()`: `GLLibPlayer`
- `Snd_Pause()`: `GLLibPlayer`
- `Snd_Play()`: `GLLibPlayer`
- `Snd_PrepareSound()`: `GLLibPlayer`
- `Snd_Go()`: `GLLibPlayer`
- `Snd_RateGet()`: `GLLibPlayer`
- `Snd_RateGetMax()`: `GLLibPlayer`
- `Snd_RateGetMin()`: `GLLibPlayer`
- `Snd_RateSet()`: `GLLibPlayer`
- `Snd_Resume()`: `GLLibPlayer`
- `Snd_SetMasterVolume()`: `GLLibPlayer`
- `Snd_SetMediaDuration()`: `GLLibPlayer`
- `Snd_Stop()`: `GLLibPlayer`
- `Snd_StopAllSounds()`: `GLLibPlayer`
- `Snd_TempoGet()`: `GLLibPlayer`
- `Snd_TempoSet()`: `GLLibPlayer`
- `Snd_UnLoadSound()`: `GLLibPlayer`
- `Snd_Update()`: `GLLibPlayer`

- `softkeyOKOnLeft`: `GLLibConfig`
- `SOLID`: `GLLib`
- `sound_enable`: `GLLibConfig`
- `sound_enableThread`: `GLLibConfig`
- `sound_numberOfChannels`: `GLLibConfig`
- `sound_useCachedPlayers`: `GLLibConfig`
- `sound_useFakeMediaDuration`: `GLLibConfig`
- `sound_useFreeChannelOnStop`: `GLLibConfig`
- `sound_useJSR135`: `GLLibConfig`
- `sound_usePrefetchedPlayers`: `GLLibConfig`
- `sound_useRealizedPlayers`: `GLLibConfig`
- `sound_useSetLevel`: `GLLibConfig`
• sound_useSetMediaTimeBeforePlay : GLLibConfig
• sound_useStopSoundsOnInterrupt : GLLibConfig
• SpecPos : GLLib
• sprite : GLLibPlayer
• sprite_alwaysBsNfm1Byte : GLLibConfig
• sprite_alwaysBsNoAfStart : GLLibConfig
• sprite_alwaysBsNoFmStart : GLLibConfig
• sprite_alwaysBsSkipFrameRc : GLLibConfig
• sprite_animFPS : GLLibConfig
• sprite_debugColision : GLLibConfig
• sprite_debugErrors : GLLibConfig
• sprite_debugLoading : GLLibConfig
• sprite_debugModuleUsage : GLLibConfig
• sprite_debugUsedMemory : GLLibConfig
• sprite_drawPixelClippingBug : GLLibConfig
• sprite_drawRegionFlippedBug : GLLibConfig
• sprite_drawRGBTransparencyBug : GLLibConfig
• sprite_fillRoundRectBug : GLLibConfig
• sprite_fontBackslashChangePalette : GLLibConfig
• sprite_fontUseOneFramePerLetter : GLLibConfig
• sprite_fpsRegion : GLLibConfig
• sprite_ModuleMapping_useModuleImages : GLLibConfig
• sprite_newTextRendering : GLLibConfig
• sprite_RGBArraysUseDrawRGB : GLLibConfig
• sprite_useAfOffShort : GLLibConfig
• sprite_useBSpriteFlags : GLLibConfig
• sprite_useBugFixImageOddSize : GLLibConfig
• sprite_useCacheFlipXY : GLLibConfig
• sprite_useCachePool : GLLibConfig
• sprite_useCacheRGBArrays : GLLibConfig
• sprite_useCreateRGB : GLLibConfig
• sprite_useCreateRGBTransparencyBug : GLLibConfig
• sprite_useDeactivateSystemGc : GLLibConfig
• sprite_useDrawRegionClipping : GLLibConfig
• sprite_useDrawStringSleep : GLLibConfig
• sprite_useDynamicPng : GLLibConfig
• sprite_useEncodeFormatI127RLE : GLLibConfig
• sprite_useEncodeFormatI16 : GLLibConfig
• sprite_useEncodeFormatI2 : GLLibConfig
• sprite_useEncodeFormatI256 : GLLibConfig
• sprite_useEncodeFormatI256RLE : GLLibConfig
• sprite_useEncodeFormatI4 : GLLibConfig
• sprite_useEncodeFormatI64RLE : GLLibConfig
• sprite_useExternImage : GLLibConfig
• sprite_useFMOffShort : GLLibConfig
• sprite_useFMPalette : GLLibConfig
• sprite_useFrameCollRC : GLLibConfig
• sprite_useFrameRects : GLLibConfig
• sprite_useGenPalette : GLLibConfig
• sprite_useGifHeader : GLLibConfig
• sprite_useHyperFM : GLLibConfig
• sprite_useIndexExAframes : GLLibConfig
• sprite_useIndexExFmodules : GLLibConfig
• sprite_useLoadImageWithoutTransf : GLLibConfig
• sprite_useModuleColorAsByte : GLLibConfig
• sprite_useModuleDataOffAsShort : GLLibConfig
• sprite_useModuleMapping : GLLibConfig
• sprite_useModuleUsageFromSprite : GLLibConfig
• sprite_useModuleWHSShort : GLLibConfig
• sprite_useModuleXY : GLLibConfig
• sprite_useModuleXYShort : GLLibConfig
• sprite_useMultipleModuleTypes : GLLibConfig
• sprite_useNokia7650DrawPixelBug : GLLibConfig
• sprite_useNokiaUI : GLLibConfig
• sprite_useNonInterlaced : GLLibConfig
• sprite_useOperationMark : GLLibConfig
• sprite_useOperationRecord : GLLibConfig
• sprite_useOperationRect : GLLibConfig
• sprite_usePixelFormat0565 : GLLibConfig
• sprite_usePixelFormat1555 : GLLibConfig
• sprite_usePixelFormat4444 : GLLibConfig
• sprite_usePixelFormat8888 : GLLibConfig
• sprite_usePrecomputedCRC : GLLibConfig
• sprite_usePrecomputedFrameRect : GLLibConfig
• sprite_useResize : GLLibConfig
• sprite_useSingleArrayForFMAF : GLLibConfig
• sprite_useSingleDirectGraphics : GLLibConfig
• sprite_useSingleImageForAllModules : GLLibConfig
• sprite_useSkipFastVisibilityTest : GLLibConfig
• sprite_useTransfFlip : GLLibConfig
• sprite_useTransfRot : GLLibConfig
• sprite_useTruncatedRGBBuffer : GLLibConfig
• Stream_Read() : GLLib
• Stream_Read16() : GLLib
• Stream_Read32() : GLLib
• Stream_ReadFully() : GLLib
• StringTokenize() : ASprite
• SUPPORTED_VERSION : ASprite

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
Here is a list of all class members with links to the classes they belong to:

- **TCP**
  - temp_byte: **ASprite**
  - temp_int: **ASprite**
  - temp_short: **ASprite**
  - Text_BuildStringCache(): **GLLib**
  - text_encoding: **GLLib**
  - Text_FreeAll(): **GLLib**
  - Text_FromUTF8(): **GLLib**
  - Text_GetLanguageAsString(): **GLLib**
  - Text_GetNbString(): **GLLib**
  - Text_GetPhoneDefaultLangage(): **GLLib**
  - Text_GetString(): **GLLib**
  - Text_LoadTextFromPack(): **GLLib**
  - text_nbString: **GLLib**
  - Text_SetEncoding(): **GLLib**
  - text_useInternalUTF8Converter: **GLLibConfig**
  - text_useStringCache: **GLLibConfig**
  - TextFitToFixedWidth(): **ASprite**
  - Tileset_Destroy(): **GLLibPlayer**
  - Tileset_Draw(): **GLLibPlayer**
  - Tileset_GetBufferGraphics(): **GLLibPlayer**
  - Tileset_GetBufferImage(): **GLLibPlayer**
  - Tileset_GetCamera(): **GLLibPlayer**
  - Tileset_GetCameraX(): **GLLibPlayer**
  - Tileset_GetCameraY(): **GLLibPlayer**
  - Tileset_GetLayerHeight(): **GLLibPlayer**
  - Tileset_GetLayerTileCountHeight(): **GLLibPlayer**
  - Tileset_GetLayerTileCountWidth(): **GLLibPlayer**
  - Tileset_GetLayerWidth(): **GLLibPlayer**
- Tileset_GetTile(): `GLLibPlayer`
- Tileset_GetTileFlags(): `GLLibPlayer`
- Tileset_Init(): `GLLibPlayer`
- Tileset_LoadLayer(): `GLLibPlayer`
- tileset_maxLayerCount: `GLLibConfig`
- Tileset_SetCamera(): `GLLibPlayer`
- Tileset_Update(): `GLLibPlayer`
- tileset_useIndexAsShort: `GLLibConfig`
- tileset_useTileShift: `GLLibConfig`
- tkNumMoveBits: `GLLib`
- TMP_BUFFER_SIZE: `GLLibConfig`
- TOP: `GRPH`, `GLLib`
- TR: `GLLang`
- TRANS_MIRROR: `GLLib`
- TRANS_MIRROR_ROT180: `GLLib`
- TRANS_MIRROR_ROT270: `GLLib`
- TRANS_MIRROR_ROT90: `GLLib`
- TRANS_NONE: `GLLib`
- TRANS_ROT180: `GLLib`
- TRANS_ROT270: `GLLib`
- TRANS_ROT90: `GLLib`
- transform_int: `ASprite`
- TransformRGB(): `ASprite`
- Translate(): `ASprite`
- tRNS: `ASprite`
Here is a list of all class members with links to the classes they belong to:

- u -

- UnInit() : GLLib
- unload() : ASprite
- Update() : GLLibPlayer
- UpdateNumberSize() : ASprite
- UpdateStringOrCharsSize() : ASprite
- UpdateStringSize() : ASprite
- upsubid : HTTP
- url : TCP
- useAbsoluteValueOfKeyCode : GLLibConfig
- useBugFixMultipleKeyPressed : GLLibConfig
- useCallSerially : GLLibConfig
- useDrawLineClippingBug : GLLibConfig
- useDrawPartialRGB : GLLibConfig
- useFakeInterruptHandling : GLLibConfig
- useFlashLightInsteadOfVibration : GLLibConfig
- useFrameDT : GLLibConfig
- useKeyAccumulation : GLLibConfig
- userAgent : HTTP
- useSafeDrawRegion : GLLibConfig
- useSafeFillRect : GLLibConfig
- useServiceRepaints : GLLibConfig
- useSleepInsteadOfYield : GLLibConfig
- useSoftwareDoubleBuffer : GLLibConfig
- useSystemGc : GLLibConfig
Here is a list of all class members with links to the classes they belong to:

- **V** -

- VCENTER : [GLLib](#), [GRPH](#)
- Vibrate() : [GLLib](#)
Here is a list of all class members with links to the classes they belong to:

- W -

- Warning() : GLLib
- WasAnyKeyPressed() : GLLib
- WasAnyKeyReleased() : GLLib
- WasKeyPressed() : GLLib
- WasKeyReleased() : GLLib
- WRAP_CLAMP : GLLibPlayer
- WRAP_REPEAT : GLLibPlayer
- WraptexB() : ASprite
- wRef : ASprite
- wTarget : ASprite
Here is a list of all class members with links to the classes they belong to:

- x -

- x_up_calling_line_id : HTTP
- x_up_subno : HTTP
- XPlayer() : XPlayer
- xplayer_CARRIER_MXTELCEL : GLLibConfig
- xplayer_CARRIER_USCINGULAR_BLUE : GLLibConfig
- xplayer_CARRIER_USCINGULAR_ORANGE : GLLibConfig
- xplayer_CARRIER_USNEXTEL : GLLibConfig
- xplayer_CARRIER_USSPRINT : GLLibConfig
- xplayer_CARRIER_USVIRGIN : GLLibConfig
- xplayer_CONN_TIMEOUT : GLLibConfig
- xplayer_ENABLE_DEBUG : GLLibConfig
- xplayer_ENABLE_DUAL_TCP : GLLibConfig
- xplayer_ENABLE_FIND_PLAYER : GLLibConfig
- xplayer_ENABLE_M7_SUPPORT : GLLibConfig
- xplayer_ENABLE_MULTIPLAYER : GLLibConfig
- xplayer_ENABLE_PLAYER_SPECIFIC_DATA : GLLibConfig
- xplayer_ENABLE_TIMEOUT : GLLibConfig
- xplayer_HTTP_NO_CANCEL : GLLibConfig
- xplayer_KEEP_ALIVE_TIME : GLLibConfig
- xplayer_USE_BUG_FIX_MESSAGE_SIZE : GLLibConfig
- xplayer_USE_HTTP_POST : GLLibConfig
- xplayer_XPLAYER_VERSION : GLLibConfig
- xRatio : ASprite
Here is a list of all class members with links to the classes they belong to:

- y -

  • yRatio : ASprite
- a -

- addByte() : XPlayer
- addByteArray() : XPlayer
- addInt() : XPlayer
- addMultipleScoreEntry() : XPlayer
- addMultipleScoreEntryWithSupplementalData() : XPlayer
- addShort() : XPlayer
- addString() : XPlayer
- ASprite() : ASprite
- Assert() : GLLib
- b -

- BuildAnimCacheImages() : ASprite
- BuildCacheImages() : ASprite
- BuildFrameCacheImages() : ASprite
- C -

- cancel() : HTTP, License, XPlayer
- CheckAndDumpConfig() : GLLib
- cleanup() : HTTP, License, XPlayer
- clearData() : XPlayer
- ClipRect() : GLLib
- connect() : TCP
- ConvertFixedPointToString() : GLLib
- CopyArea() : GLLib
- CountFrameModules() : ASprite
- Crc32() : ASprite
- d -

- Dbg() : **GLLib**
- DecodeImage() : **ASprite**
- DecodeImageToByteArray() : **ASprite**
- disconnect() : **TCP**
- DrawArc() : **GLLib**
- DrawChar() : **GLLib**
- DrawChars() : **GLLib**
- DrawImage() : **GLLib**
- DrawLine() : **ASprite** , **GLLib**
- DrawNumber() : **ASprite**
- DrawPage() : **ASprite**
- DrawPageB() : **ASprite**
- drawPartialRGB() : **GLLib**
- DrawRect() : **GLLib**
- DrawRegion() : **GLLib**
- DrawRGB() : **GLLib**
- DrawRoundRect() : **GLLib**
- DrawString() : **ASprite** , **GLLib** , **ASprite**
- DrawStringOrChars() : **ASprite**
- DrawSubstring() : **GLLib**
- f -

- FillArc() : **GLLib**
- FillRect() : **GLLib**
- FillRoundRect() : **GLLib**
- FillTriangle() : **GLLib**
- FreeAnimCacheImages() : **ASprite**
- FreeCacheData() : **ASprite**
- FreeFrameCacheImages() : **ASprite**
- FreeMemory() : **ASprite**
- FreeModuleImage() : **ASprite**
- g -

- Game_KeyClearKeyCode() : GLLib
- Game_KeySetKeyCode() : GLLib
- Game_Run() : GLLib
- Game_update() : GLLib
- Gc() : GLLib
- GenPalette() : ASprite
- GetAFrameFlags() : ASprite
- GetAFrameRect() : ASprite
- GetAFrames() : ASprite
- GetAFramesOX() : ASprite
- GetAFramesOY() : ASprite
- GetAFrameTime() : ASprite
- GetAnim() : GLLibPlayer
- GetAnimFrame() : ASprite
- GetBlueComponent() : GLLib
- GetBold() : ASprite
- getByte() : XPlayer
- getByteArray() : XPlayer
- getByteDataList() : XPlayer
- GetCharFrame() : ASprite
- GetCharMap() : ASprite
- GetCharMapShort() : ASprite
- GetChars() : ASprite
- getCharSize() : ASprite
- GetCharSpacing() : ASprite
- GetClipHeight() : GLLib
- GetClipWidth() : GLLib
- GetClipX() : GLLib
- GetClipY() : GLLib
- GetColor() : GLLib
GetCurrentMMapping() : ASprite
GetCurrentPalette() : ASprite
getCurrentPlayerLeaderboardPosition() : XPlayer
getCurrentPlayerLeaderboardScore() : XPlayer
getCurrentPlayerLeaderboardScoreData() : XPlayer
GetCurrentStringHeight() : ASprite
GetCurrentStringLength() : ASprite
getData() : XPlayer
getDataList() : XPlayer
GetDisplayColor() : GLLib
GetDuration() : GLLibPlayer
getFirstSessionIndex() : XPlayer
GetFModuleOX() : ASprite
GetFModuleOY() : ASprite
GetFModuleRect() : ASprite
GetFModules() : ASprite
GetFont() : GLLib
GetFontHeight() : ASprite
getFoundPlayerName() : XPlayer
getFoundPlayerSessionName() : XPlayer
getFoundPlayerSessionNumberOfPlayers() : XPlayer
getFoundPlayerStatus() : XPlayer
GetFrame() : GLLibPlayer
GetFrameCount() : ASprite
GetFrameHeight() : ASprite
GetFrameMarkers() : ASprite
GetFrameModule() : ASprite
GetFrameModuleFlags() : ASprite
GetFrameModuleHeight() : ASprite
GetFrameModulePalette() : ASprite
GetFrameModuleWidth() : ASprite
GetFrameModuleX() : ASprite
GetFrameModuleY() : ASprite
GetFrameRect() : ASprite
GetFrameRectCount() : ASprite
GetFrames() : ASprite
GetFrameTime() : GLLib
GetFrameWidth() : ASprite
GetGrayScale() : GLLib
- GetGreenComponent() : **GLLib**
- getInt() : **XPlayer**
- getLastError() : **License**, **XPlayer**
- getLeaderboardData() : **XPlayer**
- getLeaderboardEntryPlayerName() : **XPlayer**
- getLeaderboardEntryPlayerPosition() : **XPlayer**
- getLeaderboardEntryPlayerScore() : **XPlayer**
- getLeaderboardEntryPlayerScoreData() : **XPlayer**
- getLeaderboardSize() : **XPlayer**
- getLength() : **XPlayer**
- GetLineHeight() : **ASprite**
- GetLineSpacing() : **ASprite**
- GetMIME() : **GLLib**
- GetModuleCount() : **ASprite**
- GetModuleData() : **ASprite**
- GetModuleHeight() : **ASprite**
- GetModuleHeightOrg() : **ASprite**
- GetModuleImage() : **ASprite**
- GetModuleRect() : **ASprite**
- GetModuleWidth() : **ASprite**
- GetModuleWidthOrg() : **ASprite**
- GetModuleX() : **ASprite**
- GetModuleY() : **ASprite**
- getMyAvgScore() : **XPlayer**
- getMyBestRank() : **XPlayer**
- getMyHighScore() : **XPlayer**
- getMyHighScoreData() : **XPlayer**
- getMyLastTimePlayed() : **XPlayer**
- getMyLowScore() : **XPlayer**
- getMyLowScoreData() : **XPlayer**
- getMyNumberOfGamesPlayed() : **XPlayer**
- getNameList() : **XPlayer**
- GetNbanim() : **GLLibPlayer**
- GetNBData() : **GLLib**
- GetNbFrame() : **GLLibPlayer**
- getNewRankAfterScoreSending() : **XPlayer**
- getNumberOfItems() : **XPlayer**
- GetPalette() : **ASprite**
- getPlayerStats() : **XPlayer**
- GetRealTime(): **GLLib**
- GetRedComponent(): **GLLib**
- getRequestedPlayerData(): **XPlayer**
- getRequestedPlayerNickname(): **XPlayer**
- GetResourceAsStream(): **GLLib**
- GetScreenHeight(): **GLLib**
- GetScreenWidth(): **GLLib**
- getSessionData(): **XPlayer**
- getSessionName(): **XPlayer**
- getShort(): **XPlayer**
- GetSoftwareDoubleBuffer(): **GLLib**
- GetSoftwareDoubleBufferGraphics(): **GLLib**
- GetSpaceWidth(): **ASprite**
- GetSprite(): **GLLibPlayer**
- getString(): **XPlayer**
- GetStrokeStyle(): **GLLib**
- GetTransform(): **GLLibPlayer**
- GetTranslateX(): **GLLib**
- GetTranslateY(): **GLLib**
- GetUnderline(): **ASprite**
- getUsername(): **XPlayer**
- GLLib(): **GLLib**
- GLLibMidiSpectrumAnalyzer(): **GLLibMidiSpectrumAnalyzer**
- GLLibPathFinding(): **GLLibPathFinding**
- GLLibPlayer(): **GLLibPlayer**
- h -

- handleChangeUsername() : XPlayer
- handleHighscore() : XPlayer
- handleLogin() : XPlayer
- handleRankGet() : XPlayer
- handleRankGetAroundPlayer() : XPlayer
- handleRateGame() : XPlayer
- handleRecommendGame() : XPlayer
- handleStatsGet() : XPlayer
- handleValidateLicense() : License
- hideNotify() : GLLib
- i -

- Init() : **GLLib**
- InitCachePool() : **ASprite**
- initMultipleScores() : **XPlayer**
- InitPoolSize() : **ASprite**
- InitSharedRms() : **GLLib**
- IsAnimOver() : **GLLibPlayer**
- IsAnyKeyDown() : **GLLib**
- isErrorOccurred() : **HTTP**
- isInProgress() : **HTTP**
- IsKeyDown() : **GLLib**
- IsKeyUp() : **GLLib**
- isLicenseValid() : **License**
- isLoggedIn() : **XPlayer**

*Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2*
- k -

- keyPressed() : GLLib
- keyReleased() : GLLib
- I -

- License() : License
- Load() : ASprite
- LZMA_Inflate() : GLLib
- m -

- Math_Abs(): **GLLib**
- Math_Atan(): **GLLib**
- Math_AtanSlow(): **GLLib**
- Math_Bezier2D(): **GLLib**
- Math_Bezier3D(): **GLLib**
- Math_Cos(): **GLLib**
- Math_DegreeToFixedPointAngle(): **GLLib**
- Math_Det(): **GLLib**
- Math_DistPointLine(): **GLLib**
- Math_Div10(): **GLLib**
- Math_DotProduct(): **GLLib**
- Math_FixedPoint_Add(): **GLLib**
- Math_FixedPoint_DegreesToAngleFixedPoint(): **GLLib**
- Math_FixedPoint_DegreesToRadians(): **GLLib**
- Math_FixedPoint_Det(): **GLLib**
- Math_FixedPoint_Divide(): **GLLib**
- Math_FixedPoint_DotProduct(): **GLLib**
- Math_FixedPoint_LineCircleIntersect(): **GLLib**
- Math_FixedPoint_LineRectangleIntersect(): **GLLib**
- Math_FixedPoint_Multiply(): **GLLib**
- Math_FixedPoint_Norm(): **GLLib**
- Math_FixedPoint_NormPow(): **GLLib**
- Math_FixedPoint_PointLineDistance(): **GLLib**
- Math_FixedPoint_RadiansToAngleFixedPoint(): **GLLib**
- Math_FixedPoint_RadiansToDegrees(): **GLLib**
- Math_FixedPoint_Round(): **GLLib**
- Math_FixedPoint_Sqrt(): **GLLib**
- Math_FixedPoint_Square(): **GLLib**
- Math_FixedPoint_Subtract(): **GLLib**
- Math_FixedPointAdjust(): **GLLib**
Math_FixedPointAngleToDegree() : GLLib
Math_FixedPointToInt() : GLLib
Math_Init() : GLLib
Math_IntToFixedPoint() : GLLib
Math_Log2() : GLLib
Math_Max() : GLLib
Math_Min() : GLLib
Math_Norm() : GLLib
Math_NormPow() : GLLib
Math_QuickSort() : GLLib
Math_QuickSortIndices() : GLLib
Math_Quit() : GLLib
Math_Rand() : GLLib
Math_RandSetSeed() : GLLib
Math_RectIntersect() : GLLib
Math_SameSign() : GLLib
Math_SegmentIntersect() : GLLib
Math_Sin() : GLLib
Math_Sqrt() : GLLib
Math_Sqrt_FixedPoint() : GLLib
Math_Tan() : GLLib
Mem_ArrayCopy() : GLLib
Mem_GetArray() : GLLib
Mem_GetByte() : GLLib
Mem_GetInt() : GLLib
Mem_GetLong() : GLLib
Mem_GetShort() : GLLib
Mem_ReadArray() : GLLib
Mem_SetArray() : GLLib
Mem_SetByte() : GLLib
Mem_SetInt() : GLLib
Mem_SetLong() : GLLib
Mem_SetShort() : GLLib
MidiSpectrumAnalyzer_Parse() : GLLibMidiSpectrumAnalyzer
MidiSpectrumAnalyzer_Reset() : GLLibMidiSpectrumAnalyzer
MidiSpectrumAnalyzer_Update() : GLLibMidiSpectrumAnalyzer
ModifyPalette() : ASprite
ModifyPaletteAlpha() : ASprite
ModifyPaletteAlphaUsingAltPalette() : ASprite
- ModifyPaletteAlphaUsingLastPalette() : ASprite
- mpDisconnect() : XPlayer
- mpHandleCreateSession() : XPlayer
- mpHandleDisconnect() : XPlayer
- mpHandleEstablishConnection() : XPlayer
- mpHandleFindPlayer() : XPlayer
- mpHandleFinishGame() : XPlayer
- mpHandleGameData() : XPlayer
- mpHandleGetPlayerData() : XPlayer
- mpHandleJoinSession() : XPlayer
- mpHandleKickOutPlayer() : XPlayer
- mpHandleLeaveSession() : XPlayer
- mpHandleListSession() : XPlayer
- mpHandleLogin() : XPlayer
- mpHandleStartGame() : XPlayer
- mpHandleUpdates() : XPlayer
- mpHasOpponentFinished() : XPlayer
- mpIsConnected() : XPlayer
- mpIsInGame() : XPlayer
- mpIsInSession() : XPlayer
- mpIsLoggedIn() : XPlayer
- mpIsMaster() : XPlayer
- mpPrepareGameData() : XPlayer
- mpSendCreateSession() : XPlayer
- mpSendDisconnect() : XPlayer
- mpSendEstablishConnection() : XPlayer
- mpSendFindPlayer() : XPlayer
- mpSendFinishGame() : XPlayer
- mpSendGameData() : XPlayer
- mpSendGetPlayerData() : XPlayer
- mpSendJoinSession() : XPlayer
- mpSendKickOutPlayer() : XPlayer
- mpSendLeaveSession() : XPlayer
- mpSendListSession() : XPlayer
- mpSendLogin() : XPlayer
- mpSendQuickGame() : XPlayer
- mpSendStartGame() : XPlayer
- p -

- Pack_Close() : **GLLib**
- Pack_CloseShared() : **GLLib**
- Pack_LoadMIME() : **GLLib**
- Pack_Open() : **GLLib**
- Pack_OpenShared() : **GLLib**
- Pack_PositionAtData() : **GLLib**
- Pack_Read() : **GLLib**
- Pack_Read16() : **GLLib**
- Pack_Read32() : **GLLib**
- Pack_ReadArray() : **GLLib**
- Pack_ReadData() : **GLLib**
- Pack_ReadFully() : **GLLib**
- Pack_ReleaseBinaryCache() : **GLLib**
- Pack_Seek() : **GLLib**
- Pack_Skip() : **GLLib**
- paint() : **GLLib**
- PaintAFrame() : **ASprite**
- PaintFModule() : **ASprite**
- PaintFrame() : **ASprite**
- PaintModule() : **ASprite**
- PaintPrecomputedFrame() : **ASprite**
- PathFinding_Exec() : **GLLibPathFinding**
- PathFinding_Free() : **GLLibPathFinding**
- PathFinding_GetPathLength() : **GLLibPathFinding**
- PathFinding_GetPathPosition() : **GLLibPathFinding**
- PathFinding_GetPathPositionX() : **GLLibPathFinding**
- PathFinding_GetPathPositionY() : **GLLibPathFinding**
- PathFinding_Init() : **GLLibPathFinding**
- Pause() : **GLLib**
- PlatformRequest() : **GLLib**
• PrecomputeAllFrames() : ASprite
• PrecomputeFrame() : ASprite
• Print() : GLLib
• Profiler_BeginNamedEvent() : GLLib
• Profiler_Draw() : GLLib
• Profiler_End() : GLLib
• Profiler_EndNamedEvent() : GLLib
• Profiler_Start() : GLLib
- q -

- Quit() : GLLib
- r -

- recvPacket() : TCP
- Render() : GLLibPlayer
- Reset() : GLLibPlayer
- ResetAKey() : GLLib
- ResetCachePool() : ASprite
- ResetKey() : GLLib
- ResizeCoords() : ASprite
- Resume() : GLLib
- Rms_Read() : GLLib
- Rms_ReadShared() : GLLib
- Rms_Write() : GLLib
- Rms_WriteShared() : GLLib
- run() : HTTP, GLLib, GLLibPlayer, TCP
- S -

- SavePack() : GLLib
- scaleX() : ASprite
- scaleY() : ASprite
- sendByGet() : HTTP
- sendChangeUsername() : XPlayer
- sendEstablishConnectionPackageOnReceive() : TCP
- sendHighscore() : XPlayer
- sendHighscoreWithSupplementalData() : XPlayer
- sendLogin() : XPlayer
- sendMultipleHighscores() : XPlayer
- sendPacket() : TCP
- sendRankGet() : XPlayer
- sendRankGetAroundPlayer() : XPlayer
- sendRateGame() : XPlayer
- sendRecommendGame() : XPlayer
- sendStatsGet() : XPlayer
- sendValidateLicense() : License
- SetAnim() : GLLibPlayer
- SetBold() : ASprite
- SetCharMap() : ASprite
- SetCharMapStatic() : ASprite
- SetCharSpacing() : ASprite
- SetCharSpacingToDefault() : ASprite
- SetClip() : GLLib
- SetColor() : GLLib
- setColor() : GLLib
- SetCurrentGraphics() : GLLib
- SetCurrentMMapping() : ASprite
- SetCurrentPalette() : ASprite
- setData() : XPlayer
- SetDefaultFontMetrics() : ASprite
- SetFont() : GLLib
- SetFrame() : GLLibPlayer
- SetGraphics() : ASprite
- SetGrayScale() : GLLib
- SetLineHeight() : ASprite
- SetLineHeightToDefault() : ASprite
- SetLineSpacing() : ASprite
- SetLineSpacingToDefault() : ASprite
- SetModuleImage() : ASprite
- SetModuleImagesArray() : ASprite
- SetModuleMapping() : ASprite
- setPhoneNumber() : XPlayer
- SetPool() : ASprite
- SetPos() : GLLibPlayer
- SetResizeParameters() : ASprite
- SetSpaceWidth() : ASprite
- SetSpaceWidthToDefault() : ASprite
- SetSprite() : GLLibPlayer
- SetStrokeStyle() : GLLib
- SetSubString() : ASprite
- SetTempBuffer() : ASprite
- SetTransform() : GLLibPlayer
- SetUnderline() : ASprite
- SetupDefaultKey() : GLLib
- SetupDisplay() : GLLib
- setUsername() : XPlayer
- showNotify() : GLLib
- sizeChanged() : GLLib
- Snd_DurationGet() : GLLibPlayer
- Snd_ForceExecOnThreadOnGamePause() : GLLibPlayer
- Snd_FreeChannel() : GLLibPlayer
- Snd_GetChannelPlayer() : GLLibPlayer
- Snd_GetChannelVolume() : GLLibPlayer
- Snd_GetCurrentSoundIndex() : GLLibPlayer
- Snd_Init() : GLLibPlayer
- Snd_IsPlaying() : GLLibPlayer
- Snd_LoadSound() : GLLibPlayer
- Snd_MediaTimeGet() : GLLibPlayer
• Snd_MediaTimeSet() : GLLibPlayer
• Snd_MidiPlayNote() : GLLibPlayer
• Snd_MidiSetChannelVolume() : GLLibPlayer
• Snd_Pause() : GLLibPlayer
• Snd_Play() : GLLibPlayer
• Snd_PrepareSound() : GLLibPlayer
• Snd_Quit() : GLLibPlayer
• Snd_RateGet() : GLLibPlayer
• Snd_RateGetMax() : GLLibPlayer
• Snd_RateGetMin() : GLLibPlayer
• Snd_RateSet() : GLLibPlayer
• Snd_Resume() : GLLibPlayer
• Snd_SetMasterVolume() : GLLibPlayer
• Snd_SetMediaDuration() : GLLibPlayer
• Snd_Stop() : GLLibPlayer
• Snd_StopAllSounds() : GLLibPlayer
• Snd_TempoGet() : GLLibPlayer
• Snd_TempoSet() : GLLibPlayer
• Snd_UnLoadSound() : GLLibPlayer
• Snd_Update() : GLLibPlayer
• Stream_Read() : GLLib
• Stream_Read16() : GLLib
• Stream_Read32() : GLLib
• Stream_ReadFully() : GLLib
• StringTokenize() : ASprite
- t -

- TCP() : TCP
  - Text_BuildStringCache() : GLLib
  - Text_FreeAll() : GLLib
  - Text_FromUTF8() : GLLib
  - Text_GetLanguageAsString() : GLLib
  - Text_GetNbString() : GLLib
  - Text_GetPhoneDefaultLangage() : GLLib
  - Text_GetString() : GLLib
  - Text_LoadTextFromPack() : GLLib
  - Text_SetEncoding() : GLLib
  - TextFitToFixedWidth() : ASprite
  - Tileset_Destroy() : GLLibPlayer
  - Tileset_Draw() : GLLibPlayer
  - Tileset_GetBufferGraphics() : GLLibPlayer
  - Tileset_GetBufferImage() : GLLibPlayer
  - Tileset_GetCamera() : GLLibPlayer
  - Tileset_GetCameraX() : GLLibPlayer
  - Tileset_GetCameraY() : GLLibPlayer
  - Tileset_GetLayerHeight() : GLLibPlayer
  - Tileset_GetLayerTileCountHeight() : GLLibPlayer
  - Tileset_GetLayerTileCountWidth() : GLLibPlayer
  - Tileset_GetLayerWidth() : GLLibPlayer
  - Tileset_GetTile() : GLLibPlayer
  - Tileset_GetTileFlags() : GLLibPlayer
  - Tileset_Init() : GLLibPlayer
  - Tileset_LoadLayer() : GLLibPlayer
  - Tileset_SetCamera() : GLLibPlayer
  - Tileset_Update() : GLLibPlayer
  - TransformRGB() : ASprite
  - Translate() : GLLib
- u -

- UnInit() : **GLLib**
- unload() : **ASprite**
- Update() : **GLLibPlayer**
- UpdateNumberSize() : **ASprite**
- UpdateStringOrCharsSize() : **ASprite**
- UpdateStringLength() : **ASprite**
- V -

- Vibrate() : GLLib
- W -

- Warning() : **GLLib**
- WasAnyKeyPressed() : **GLLib**
- WasAnyKeyReleased() : **GLLib**
- WasKeyPressed() : **GLLib**
- WasKeyReleased() : **GLLib**
- WraptextB() : **ASprite**
- X -

- XPlayer() : XPlayer
- a -

- Align : GLLib
- b -

- BASE : ASprite
- BASIC_MESSAGE_TYPE_CONNECT : MessageType, XPlayer.MessageType
- BASIC_MESSAGE_TYPE_GAME : MessageType, XPlayer.MessageType
- BASIC_MESSAGE_TYPE_UNKNOWN : XPlayer.MessageType, MessageType
- BLOCK_INFO_SIZE : ASprite
- BOTTOM : GLLib, GRPH
- BR : GLLang
- BS_AF_OFF_SHORT : ASprite
- BS_ANIMS : ASprite
- BS_DEFAULT_DOJA : ASprite
- BS_DEFAULT_MIDP1 : ASprite
- BS_DEFAULT_MIDP1b : ASprite
- BS_DEFAULT_MIDP1c : ASprite
- BS_DEFAULT_MIDP2 : ASprite
- BS_DEFAULT_NOKIA : ASprite
- BS_FM_OFF_SHORT : ASprite
- BS_FM_PALETTE : ASprite
- BS_FRAME_COLL_RC : ASprite
- BS_FRAME_RECTS : ASprite
- BS_FRAMES : ASprite
- BS_GIF_HEADER : ASprite
- BS_IMAGE_SIZE_INT : ASprite
- BS_KEEP_PAL : ASprite
- BS_MODULE_IMAGES : ASprite
- BS_MODULE_IMAGES_FX : ASprite
- BS_MODULE_USAGE : ASprite
- BS_MODULES : ASprite
- BS_MODULES_IMG: ASprite
- BS_MODULES_USAGE: ASprite
- BS_MODULES_WH_SHORT: ASprite
- BS_MODULES_XY: ASprite
- BS_MODULES_XY_SHORT: ASprite
- BS_NAF_1_BYTE: ASprite
- BS_NFM_1_BYTE: ASprite
- BS_NO_AF_START: ASprite
- BS_PNG_CRC: ASprite
- BS.Skip FRAME_RC: ASprite
- BS.TRANSP_FIRST: ASprite
- BS.TRANSP_LAST: ASprite
- BSPIRITE_v003: ASprite
- BSPIRITE_v004: ASprite
- BSPIRITE_v005: ASprite
- c -

- callstarttime : License, XPlayer  
- CarrierDeviceId : HTTP  
- clientId : HTTP  
- CN : GLLang  
- CONNECT_MESSAGE_DISCONNECT : XPlayer.MessageType, MessageType  
- CONNECT_MESSAGE_ERROR : XPlayer.MessageType, MessageType  
- CONNECT_MESSAGE_INIT_READ : XPlayer.MessageType, MessageType  
- CONNECT_MESSAGE_INIT_WRITE : MessageType, XPlayer.MessageType  
- CONNECT_MESSAGE_SUCCESS : XPlayer.MessageType, MessageType  
- connection : TCP  
- CRC32_POLYNOMIAL : ASprite  
- crcTable : ASprite  
- curFlags : GLLibPlayer  
- currentChunkType : ASprite  
- curTime : GLLibPlayer  
- CZ : GLLang
- d -

- **DE**: `GLLang`
- `disableNotifyDestroyed`: `GLLibConfig`
- `doja_Network_ErrorNoCredits`: `GLLibConfig`
- `doja_Network_ErrorNoNetwork`: `GLLibConfig`
- `doja_Network_ErrorNoNetworkAccess`: `GLLibConfig`
- `doja_Network_NoError`: `GLLibConfig`
- `doja_ScratchPad_CreditsFile`: `GLLibConfig`
- `doja_ScratchPad_EOF`: `GLLibConfig`
- `doja_ScratchPad_SaveGameFile`: `GLLibConfig`
- `DOTTED`: `GLLib`
- e -

- **EN**: **GLLang**
- ENCODE_FORMAT_I127RLE : **ASprite**
- ENCODE_FORMAT_I16 : **ASprite**
- ENCODE_FORMAT_I2 : **ASprite**
- ENCODE_FORMAT_I256 : **ASprite**
- ENCODE_FORMAT_I256RLE : **ASprite**
- ENCODE_FORMAT_I4 : **ASprite**
- ENCODE_FORMAT_I64RLE : **ASprite**
- ERROR_BAD_RESPONSE : License.Error, XPlayer.Error
- ERROR_BLOCKED_SESSION : MessageType, XPlayer.MessageType
- ERROR_CHANGE_USERNAME_FAILED : XPlayer.Error
- ERROR_CONNECTION : XPlayer.Error, License.Error
- ERROR_CREATE_SESSION_INVALID_NAME : XPlayer.MessageType, MessageType
- ERROR_CREATE_SESSION_NAME_USED : XPlayer.MessageType, MessageType
- ERROR_GET_RANKINGS_FAILED : XPlayer.Error
- ERROR_GET_STATS_FAILED : XPlayer.Error
- ERROR_INIT : XPlayer.Error, License.Error
- ERROR_INVALID_FILE_LENGTH : **GLLibMidiSpectrumAnalyzer**
- ERROR_INVALID_GGI : License.Error, XPlayer.Error
- ERROR_INVALID_HEADER_CHUNK : **GLLibMidiSpectrumAnalyzer**
- ERROR_INVALID_META_EVENT : **GLLibMidiSpectrumAnalyzer**
- ERROR_INVALID_TRACK_CHUNK : **GLLibMidiSpectrumAnalyzer**
- ERROR_JOIN_GAME : XPlayer.Error
- ERROR_JOIN_SESSION_TOO_MANY_PLAYERS : MessageType
• ERROR_KICK_OUT_PLAYER_IS_MASTER : MessageType , XPlayer.MessageType
• ERROR_LIST_SESSION_INVALID_INDEX : MessageType , XPlayer.MessageType
• ERROR_LOGIN_AUTHENTICATION_FAILED : XPlayer.MessageType , MessageType
• ERROR_LOGIN_INVALID_NICKNAME : MessageType , XPlayer.MessageType
• ERROR_LOGIN_NICKNAME_USED : XPlayer.MessageType , MessageType
• ERROR_NICK_TAKEN : XPlayer.Error
• ERROR_NO_CLIENT_ID : License.Error , XPlayer.Error
• ERROR_NO_NICKNAME : XPlayer.Error
• ERROR_NO_PHONE_NUMBER : License.Error , XPlayer.Error
• ERROR_NO_PLAYER : MessageType , XPlayer.MessageType
• ERROR_NO_SESSION : XPlayer.MessageType , MessageType
• ERROR_NO_UUID : XPlayer.Error , License.Error
• ERROR_NONE : XPlayer.Error , License.Error
• ERROR_NOT_M7_ENABLED : XPlayer.Error
• ERROR_NOT_MASTER : XPlayer.MessageType , MessageType
• ERROR_NOT_REGISTERED : XPlayer.Error
• ERROR_PENDING : License.Error , XPlayer.Error
• ERROR_RATE_GAME_FAILED : XPlayer.Error
• ERROR_RECOMMEND_GAME_FAILED : XPlayer.Error
• ERROR_REGISTER_FAILED : XPlayer.Error
• ERROR_REQUEST_FAILED : XPlayer.Error
• ERROR_SCORE_UPLOAD_FAILED : XPlayer.Error
• ERROR_SESSION_CLOSED : MessageType , XPlayer.MessageType
• ERROR_START_GAME : XPlayer.Error
• ERROR_START_GAME_NOT_ENOUGH_PLAYERS : XPlayer.MessageType , MessageType
• ERROR_SUPPLEMENTAL_DATA_NEEDED : XPlayer.Error
• ERROR_VALIDATE_LICENSE_FAILED : XPlayer.Error
• ERROR_WRONGFULL_QSTATE : XPlayer.Error
• ERROR_WRONGFULL_RSTATE : XPlayer.Error
• ES : GLLang
- f -

- FakeInterruptThreshold : **GLLibConfig**
- FIXED_PRECISION : **GLLibConfig**
- FLAG_FLIP_X : **ASprite**
- FLAG_FLIP_Y : **ASprite**
- FLAG_HYPER_FM : **ASprite**
- FLAG_INDEX_EX_MASK : **ASprite**
- FLAG_OFFSET_AF : **ASprite**
- FLAG_OFFSET_FM : **ASprite**
- FLAG_ROT_90 : **ASprite**
- FLAG_USER0 : **ASprite**
- FLAG_USER1 : **ASprite**
- FPSLimiter : **GLLibConfig**
- FR : **GLLang**
- g -

- **g** : **GLLib**
- **GAME_MESSAGE_ERROR** : **MessageType**, **XPlayer.MessageType**
- **GAME_MESSAGE_IN_GAME** : **MessageType**, **XPlayer.MessageType**
- **GAME_MESSAGE_IN_GAME_TOALL** : **MessageType**, **XPlayer.MessageType**
- **GAME_MESSAGE_KEEP_ALIVE** : **MessageType**, **XPlayer.MessageType**
- **GAME_MESSAGE_PUSH** : **XPlayer.MessageType**, **MessageType**
- **GAME_MESSAGE_REQUEST** : **XPlayer.MessageType**, **MessageType**
- h -

- HCENTER : GLLib, GRPH
- HCENTER_BOTTOM : GRPH
- HCENTER_TOP : GRPH
- HCENTER_VCENTER : GRPH
- HEADER_LEVEL0_MAX_WBITS : ASprite
- hRef : ASprite
- hTarget : ASprite

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
- i -

- IDAT : ASprite
- IEND : ASprite
- IHDR : ASprite
- IN_CONNECTED : XPlayer.ConnectLevel
- IN_GAME : XPlayer.ConnectLevel
- IN_GAME_MASTER : XPlayer.ConnectLevel
- IN_SESSION : XPlayer.ConnectLevel
- IN_SESSION_MASTER : XPlayer.ConnectLevel
- incomingQueue : TCP
- INDEX_EX_MASK : ASprite
- INDEX_EX_SHIFT : ASprite
- INDEX_MASK : ASprite
- INFO32 : ASprite
- INFO8 : ASprite
- inputIndex : GLLib
- isGameMessageInQueue : XPlayer
- IsMatch : GLLib
- IsRep : GLLib
- IsRep0Long : GLLib
- IsRepG0 : GLLib
- IsRepG1 : GLLib
- IsRepG2 : GLLib
- istream : TCP
- IT : GLLang
- j -

- JP : GLLang

Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2
- k -

- k_consoleLogEnable : GLKey
- k_down : GLKey
- k_dummy : GLKey
- k_fire : GLKey
- k_invalid : GLKey
- k_itoa_buffer_size : ASprite
- k_left : GLKey
- k_menuBack : GLKey
- k_menuOK : GLKey
- k_nbKey : GLKey
- k_num0 : GLKey
- k_num1 : GLKey
- k_num2 : GLKey
- k_num3 : GLKey
- k_num4 : GLKey
- k_num5 : GLKey
- k_num6 : GLKey
- k_num7 : GLKey
- k_num8 : GLKey
- k_num9 : GLKey
- k_pound : GLKey
- k_right : GLKey
- k_rmsLogEnable : GLKey
- k_snd_priority_highest : GLLibPlayer
- k_snd_priority_lowest : GLLibPlayer
- k_snd_priority_normal : GLLibPlayer
- k_star : GLKey
- k_up : GLKey
- kAlignTableSize : GLLib
- kBitModelTotal : GLLib
- `kDirDown`: `GLLibPathFinding`
- `kDirDownLeft`: `GLLibPathFinding`
- `kDirDownRight`: `GLLibPathFinding`
- `kDirLeft`: `GLLibPathFinding`
- `kDirRight`: `GLLibPathFinding`
- `kDirUp`: `GLLibPathFinding`
- `kDirUpLeft`: `GLLibPathFinding`
- `kDirUpRight`: `GLLibPathFinding`
- `kEndPosModelIndex`: `GLLib`
- `keycodeDown`: `GLLibConfig`
- `keycodeFire`: `GLLibConfig`
- `keycodeLeft`: `GLLibConfig`
- `keycodeLeftSoftkey`: `GLLibConfig`
- `keycodeRight`: `GLLibConfig`
- `keycodeRightSoftkey`: `GLLibConfig`
- `keycodeUp`: `GLLibConfig`
- `kLenNumHighBits`: `GLLib`
- `kLenNumHighSymbols`: `GLLib`
- `kLenNumLowBits`: `GLLib`
- `kLenNumLowSymbols`: `GLLib`
- `kLenNumMidBits`: `GLLib`
- `kLenNumMidSymbols`: `GLLib`
- `kMatchMinLen`: `GLLib`
- `kNumAlignBits`: `GLLib`
- `kNumBitModelTotalBits`: `GLLib`
- `kNumFullDistances`: `GLLib`
- `kNumLenProbs`: `GLLib`
- `kNumLenToPosStates`: `GLLib`
- `kNumMoveBits`: `GLLib`
- `kNumPosBitsMax`: `GLLib`
- `kNumPosSlotBits`: `GLLib`
- `kNumPosStatesMax`: `GLLib`
- `kNumStates`: `GLLib`
- `kNumTopBits`: `GLLib`
- `KR`: `GLLang`
- `kStartPosModelIndex`: `GLLib`
- `kTopValue`: `GLLib`
LEFT : GLLib, GRPH
LEFT_BOTTOM : GRPH
LEFT_TOP : GRPH
LEFT_VCENTER : GRPH
LenChoice : GLLib
LenChoice2 : GLLib
LenCoder : GLLib
LenHigh : GLLib
LenLow : GLLib
LenMid : GLLib
Literal : GLLib
lowMemoryLimit : GLLibConfig
LZMA_BASE_SIZE : GLLib
LZMA_LIT_SIZE : GLLib
LZMA_RESULT_DATA_ERROR : GLLib
LZMA_RESULT_NOT_ENOUGH_MEM : GLLib
LZMA_RESULT_OK : GLLib
- m -

- m_barLevels : **GLLibMidiSpectrumAnalyzer**
- m_bars : **GLLibMidiSpectrumAnalyzer**
- m_bError : **HTTP**, **TCP**
- m_bErrorString : **TCP**
- m_Buffer : **GLLib**
- m_channelFilter : **GLLibMidiSpectrumAnalyzer**
- m_Code : **GLLib**
- m_connected : **TCP**
- m_current_keys_pressed : **GLLib**
- m_current_keys_released : **GLLib**
- m_current_keys_state : **GLLib**
- m_currentTime : **GLLibMidiSpectrumAnalyzer**
- m_ExtraBytes : **GLLib**
- m_inSize : **GLLib**
- m_keys_pressed : **GLLib**
- m_keys_released : **GLLib**
- m_keys_state : **GLLib**
- m_last_key_pressed : **GLLib**
- m_outStream : **GLLib**
- m_Range : **GLLib**
- m_response : **HTTP**
- m_responseByteArray : **HTTP**
- MAGIC : **ASprite**
- MAGIC_IDAT_h : **ASprite**
- MAGIC_IEND : **ASprite**
- Math_Angle180 : **GLLib**
- Math_Angle270 : **GLLib**
- Math_Angle360 : **GLLib**
- Math_Angle90 : **GLLib**
- math_angleFixedPointBase : **GLLibConfig**
• Math_AngleMUL : \texttt{GLLib}
• math_AtanUseCacheTable : \texttt{GLLibConfig}
• Math_FixedPoint_E : \texttt{GLLib}
• Math_FixedPoint_PI : \texttt{GLLib}
• math_fixedPointBase : \texttt{GLLibConfig}
• MATH_INTERSECT_NO_INTERSECT : \texttt{GLLib}
• MATH_INTERSECT_ONE_POINT : \texttt{GLLib}
• MATH_INTERSECT_TWO_POINTS : \texttt{GLLib}
• MATH_SEGMENTINTERSECT_COLLINEAR : \texttt{GLLib}
• MATH_SEGMENTINTERSECT_DO_INTERSECT : \texttt{GLLib}
• MATH_SEGMENTINTERSECT_DONT_INTERSECT : \texttt{GLLib}
• MAX_FLIP_COUNT : \texttt{GLLibConfig}
• MAX_MODULE_MAPPINGS : \texttt{GLLibConfig}
• MAX_SPRITE_PALETTE : \texttt{GLLibConfig}
• MAX_TRANSFORM_FLAGS : \texttt{ASprite}
• MAX_WRAP_INFO : \texttt{GLLibConfig}
• MD_ARC : \texttt{ASprite}
• MD_FILL_ARC : \texttt{ASprite}
• MD_FILL_RECT : \texttt{ASprite}
• MD_FILL_TRIANGLE : \texttt{ASprite}
• MD_IMAGE : \texttt{ASprite}
• MD_MARKER : \texttt{ASprite}
• MD_RECT : \texttt{ASprite}
• MD_TRIANGLE : \texttt{ASprite}
• mem : \texttt{ASprite}
• midp2_flags : \texttt{ASprite}
• MIDP2forceNonFullScreen : \texttt{GLLibConfig}
• MIME_type : \texttt{GLLib}
• mod : \texttt{ASprite}
• mResizeCorrectY : \texttt{ASprite}
• mResizeRef : \texttt{ASprite}
- n -

- NMAX : **ASprite**
- NO_ERRORS : **GLLibMidiSpectrumAnalyzer**
- NOT_A_NUMBER : **XPlayer, License**
- NOT_CONNECTED : **XPlayer.ConnectLevel**
- NOT_LOGGED_IN : **XPlayer.ConnectLevel**

*Generated on Tue Sep 23 23:05:31 2008 for GLib by doxygen 1.5.2*
- O -

- OPERATION_COMPUTERECT : ASprite
- OPERATION_DRAW : ASprite
- OPERATION_MARK : ASprite
- OPERATION_RECORD : ASprite
- ostream : TCP
- outgoing_data_on_receive_connection : TCP
- outgoingQueue : TCP
- p -

- pack_dbgDataAccess : GLLibConfig
- pack_keepLoaded : GLLibConfig
- pack_skipBufferSize : GLLibConfig
- pack_supportLZMAdecompression : GLLibConfig
- pack_useBlackBerryGZipDecompression : GLLibConfig
- PAL_BLEND_BLACK : ASprite
- PAL_BLUE_CYAN : ASprite
- PAL_GREEN : ASprite
- PAL_GREY : ASprite
- PAL_INVISIBLE : ASprite
- PAL_ORIGINAL : ASprite
- PAL_RED_YELLOW : ASprite
- pathfinding_Debug : GLLibConfig
- pathfinding_MaxNode : GLLibConfig
- PIXEL_FORMAT_0565 : ASprite
- PIXEL_FORMAT_1555 : ASprite
- PIXEL_FORMAT_4444 : ASprite
- PIXEL_FORMAT_8888 : ASprite
- PL : GLLang
- platformRequestOnExit : GLLibConfig
- PLAYER_STATUS_HAS_NO_PSD : XPlayer.MessageType, MessageType
- PLAYER_STATUS_HAS_PSD : MessageType, XPlayer.MessageType
- PLAYER_STATUS_NOT_REGISTERED : MessageType, XPlayer.MessageType
- PLAYER_STATUS_OFFLINE : MessageType, XPlayer.MessageType
- PLAYER_STATUS_ONLINE : MessageType, XPlayer.MessageType
- PLAYER_STATUS_ONLINE_IN_SESSION: XPlayer.MessageType, MessageType
- PLAYER_STATUS_ONLINE_PLAYING: XPlayer.MessageType, MessageType
- PLTE: ASprite
- PNG_BUFFER_SIZE: GLLibConfig
- PNG_INFO_SIZE: ASprite
- PosSlot: GLLib
- posX: GLLibPlayer
- posY: GLLibPlayer
- PROFILER_MAX_EVENTS: GLLib
- PT: GLLang
- PUSH_MESSAGE_FINISH_GAME: MessageType, XPlayer.MessageType
- PUSH_MESSAGE_JOIN_SESSION: XPlayer.MessageType, MessageType
- PUSH_MESSAGE_KICK_OUT: MessageType, XPlayer.MessageType
- PUSH_MESSAGE_LEAVE_SESSION: MessageType, XPlayer.MessageType
- PUSH_MESSAGE_START_GAME: MessageType, XPlayer.MessageType
- r -

- receive_connection : TCP
- receive_istream : TCP
- receive_ostream : TCP
- record_frame : ASprite
- record_index : ASprite
- RepLenCoder : GLLib
- REQUEST_MESSAGE_CREATE_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_FINISH_GAME : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_PLAYER_DATA : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_PLAYER_INFO : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_GET_QUICK_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_JOIN_SESSION : MessageType , XPlayer.MessageType
- REQUEST_MESSAGE_KICK_OUT_PLAYER : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LEAVE_SESSION : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LIST_SESSIONS : XPlayer.MessageType , MessageType
- REQUEST_MESSAGE_LOGIN : MessageType , XPlayer.MessageType
- REQUEST_MESSAGE_PROXY : MessageType
- REQUEST_MESSAGE_START_GAME : XPlayer.MessageType , MessageType
- RESIZE_CREATERGB : ASprite
- RESIZE_DRAW_ON_MUTABLE : `ASprite`
- RESIZE_NONE : `ASprite`
- RESIZE_NOT_CACHED : `ASprite`
- RESIZE_REF_176x220 : `ASprite`
- RESIZE_REF_240x256 : `ASprite`
- RESIZE_REF_240x320 : `ASprite`
- RESPONSE_MESSAGE_ERROR : `XPlayer.MessageType`, `MessageType`
- RESPONSE_MESSAGE_SUCCESS : `XPlayer.MessageType`, `MessageType`
- RIGHT : `GLLib`, `GRPH`
- RIGHT_BOTTOM : `GRPH`
- RIGHT_TOP : `GRPH`
- RIGHT_VCENTER : `GRPH`
- rms_usePackRead : `GLLibConfig`
- rms_useSharing : `GLLibConfig`
- RU : `GLLang`
- S -

- s_application : **GLLib**
- s_bAspectRatio : **ASprite**
- s_bBilinear : **ASprite**
- s_data_mimeType : **GLLibPlayer**
- s_display : **GLLib**
- s_game_currentFrameNB : **GLLib**
- s_game_FPSAverage : **GLLib**
- s_game_frameDT : **GLLib**
- s_game_interruptNotify : **GLLib**
- s_game_isPaused : **GLLib**
- s_game_keyEventIndex : **GLLib**
- s_game_keyJustPressed : **GLLib**
- s_game_keyPressedTime : **GLLib**
- s_game_lastFrameTime : **GLLib**
- s_game_state : **GLLib**
- s_game_timeWhenFrameStart : **GLLib**
- s_game_totalExecutionTime : **GLLib**
- s_gllib_instance : **GLLib**
- s_keyLastKeyPressUntranslatedCode : **GLLib**
- s_keyLastKeyStates : **GLLib**
- s_keyState : **GLLib**
- s_keyStateRT : **GLLib**
- s_MapChar : **ASprite**
- s_math_bezierX : **GLLib**
- s_math_bezierY : **GLLib**
- s_math_bezierZ : **GLLib**
- s_Math_distPointLineX : **GLLib**
- s_Math_distPointLineY : **GLLib**
- s_math_F_05 : **GLLib**
- s_math_F_1 : **GLLib**
• s_Math_intersectPoints : GLLib
• s_Math_intersectX : GLLib
• s_Math_intersectY : GLLib
• s_math_random : GLLib
• s_pack_lastDataReadMimeType : GLLib
• s_rc : ASprite
• s_resizeType : ASprite
• s_snd_masterVolume : GLLibPlayer
• s_snd_maxNbSoundSlot : GLLibPlayer
• SCALE_SHIFT : ASprite
• SCORE_TYPE_POINTS : XPlayer
• SCORE_TYPE_TIME : XPlayer
• screenHeight : GLLibConfig
• screenWidth : GLLibConfig
• SLEEP_DRAWSTRINGB : GLLibConfig
• softkeyOKOnLeft : GLLibConfig
• SOLID : GLLib
• sound_enable : GLLibConfig
• sound_enableThread : GLLibConfig
• sound_numberOfChannels : GLLibConfig
• sound_useCachedPlayers : GLLibConfig
• sound_useFakeMediaDuration : GLLibConfig
• sound_useFreeChannelOnStop : GLLibConfig
• sound_useJSR135 : GLLibConfig
• sound_usePrefetchedPlayers : GLLibConfig
• sound_useRealizedPlayers : GLLibConfig
• sound_useSetLevel : GLLibConfig
• sound_useSetMediaTimeBeforePlay : GLLibConfig
• sound_useStopSoundsOnInterrupt : GLLibConfig
• SpecPos : GLLib
• sprite : GLLibPlayer
• sprite_alwaysBsNfm1Byte : GLLibConfig
• sprite_alwaysBsNoAfStart : GLLibConfig
• sprite_alwaysBsNoFmStart : GLLibConfig
• sprite_alwaysBsSkipFrameRc : GLLibConfig
• sprite_animFPS : GLLibConfig
• sprite_debugColision : GLLibConfig
• sprite_debugErrors : GLLibConfig
• sprite_debugLoading : GLLibConfig
• sprite_debugModuleUsage : GLLibConfig
• sprite_debugUsedMemory : GLLibConfig
• sprite_drawPixelClippingBug : GLLibConfig
• sprite_drawRegionFlippedBug : GLLibConfig
• sprite_drawRGBTransparencyBug : GLLibConfig
• sprite_fillRoundRectBug : GLLibConfig
• sprite_fontBackslashChangePalette : GLLibConfig
• sprite_fontUseOneFramePerLetter : GLLibConfig
• sprite_fpsRegion : GLLibConfig
• sprite_ModuleMapping_useModuleImages : GLLibConfig
• sprite_newTextRendering : GLLibConfig
• sprite_RGBArraysUseDrawRGB : GLLibConfig
• sprite_useAfOffShort : GLLibConfig
• sprite_useBSpriteFlags : GLLibConfig
• sprite_useBugFixImageOddSize : GLLibConfig
• sprite_useCacheFlipXY : GLLibConfig
• sprite_useCachePool : GLLibConfig
• sprite_useCacheRGBArrays : GLLibConfig
• sprite_useCreateRGB : GLLibConfig
• sprite_useCreateRGBTransparencyBug : GLLibConfig
• sprite_useDeactivateSystemGc : GLLibConfig
• sprite_useDrawRegionClipping : GLLibConfig
• sprite_useDrawStringSleep : GLLibConfig
• sprite_useDynamicPng : GLLibConfig
• sprite_useEncodeFormat127RLE : GLLibConfig
• sprite_useEncodeFormat16 : GLLibConfig
• sprite_useEncodeFormat2 : GLLibConfig
• sprite_useEncodeFormat256 : GLLibConfig
• sprite_useEncodeFormat256RLE : GLLibConfig
• sprite_useEncodeFormat4 : GLLibConfig
• sprite_useEncodeFormat64RLE : GLLibConfig
• sprite_useExternImage : GLLibConfig
• sprite_useFMOffShort : GLLibConfig
• sprite_useFMPalette : GLLibConfig
• sprite_useFrameColllRC : GLLibConfig
• sprite_useFrameRects : GLLibConfig
• sprite_useGenPalette : GLLibConfig
• sprite_useGifHeader : GLLibConfig
• sprite_useHyperFM : GLLibConfig
- sprite_useIndexExAframes: GLLibConfig
- sprite_useIndexExFmodules: GLLibConfig
- sprite_useLoadImageWithoutTransf: GLLibConfig
- sprite_useModuleColorAsByte: GLLibConfig
- sprite_useModuleDataOffAsShort: GLLibConfig
- sprite_useModuleMapping: GLLibConfig
- sprite_useModuleUsageFromSprite: GLLibConfig
- sprite_useModuleWHSShort: GLLibConfig
- sprite_useModuleXY: GLLibConfig
- sprite_useModuleXYShort: GLLibConfig
- sprite_useMultipleModuleTypes: GLLibConfig
- sprite_useNokia7650DrawPixelBug: GLLibConfig
- sprite_useNokiaUI: GLLibConfig
- sprite_useNonInterlaced: GLLibConfig
- sprite_useOperationMark: GLLibConfig
- sprite_useOperationRecord: GLLibConfig
- sprite_useOperationRect: GLLibConfig
- sprite_usePixelFormat0565: GLLibConfig
- sprite_usePixelFormat1555: GLLibConfig
- sprite_usePixelFormat4444: GLLibConfig
- sprite_usePixelFormat8888: GLLibConfig
- sprite_usePrecomputedCRC: GLLibConfig
- sprite_usePrecomputedFrameRect: GLLibConfig
- sprite_useResize: GLLibConfig
- sprite_useSingleArrayForFMAF: GLLibConfig
- sprite_useSingleDirectGraphics: GLLibConfig
- sprite_useSingleImageForAllModules: GLLibConfig
- sprite_useSkipFastVisibilityTest: GLLibConfig
- sprite_useTransfFlip: GLLibConfig
- sprite_useTransfRot: GLLibConfig
- sprite_useTruncatedRGBBuffer: GLLibConfig
- SUPPORTED_VERSION: ASprite
- t -

- temp_byte : ASprite
- temp_int : ASprite
- temp_short : ASprite
- text_encoding : GLLib
- text_nbString : GLLib
- text_useInternalUTF8Converter : GLLibConfig
- text_useStringCache : GLLibConfig
- tileset_maxLayerCount : GLLibConfig
- tileset_useIndexAsShort : GLLibConfig
- tileset_useTileShift : GLLibConfig
- tkNumMoveBits : GLLib
- TMP_BUFFER_SIZE : GLLibConfig
- TOP : GRPH, GLLib
- TR : GLLang
- TRANS_MIRROR : GLLib
- TRANS_MIRROR_ROT180 : GLLib
- TRANS_MIRROR_ROT270 : GLLib
- TRANS_MIRROR_ROT90 : GLLib
- TRANS_NONE : GLLib
- TRANS_ROT180 : GLLib
- TRANS_ROT270 : GLLib
- TRANS_ROT90 : GLLib
- transform_int : ASprite
- tRNS : ASprite
- u -

- upsubid : HTTP
- url : TCP
- useAbsoluteValueOfKeyCode : GLLibConfig
- useBugFixMultipleKeyPressed : GLLibConfig
- useCallSerially : GLLibConfig
- useDrawLineClippingBug : GLLibConfig
- useDrawPartialRGB : GLLibConfig
- useFakeInterruptHandling : GLLibConfig
- useFlashLightInsteadOfVibration : GLLibConfig
- useFrameDT : GLLibConfig
- useKeyAccumulation : GLLibConfig
- userAgent : HTTP
- useSafeDrawRegion : GLLibConfig
- useSafeFillRect : GLLibConfig
- useServiceRepaints : GLLibConfig
- useSleepInsteadOfYield : GLLibConfig
- useSoftwareDoubleBuffer : GLLibConfig
- useSystemGc : GLLibConfig
- V -

- VCENTER : **GLLib**, **GRPH**

*Generated on Tue Sep 23 23:05:31 2008 for GLLib by doxygen 1.5.2*
- W -

- WRAP_CLAMP : **GLLibPlayer**
- WRAP_REPEAT : **GLLibPlayer**
- wRef : **ASprite**
- wTarget : **ASprite**

*Generated on Tue Sep 23 23:05:31 2008 for GLLib by [doxygen](http://www.stackoverflow.com) 1.5.2*
- x -

- x_up_calling_line_id : HTTP
- x_up_subno : HTTP
- xplayer_CARRIER_MXTELCEL : GLLibConfig
- xplayer_CARRIER_USCINGULAR_BLUE : GLLibConfig
- xplayer_CARRIER_USCINGULAR_ORANGE : GLLibConfig
- xplayer_CARRIER_USNEXTEL : GLLibConfig
- xplayer_CARRIER_USPRINT : GLLibConfig
- xplayer_CARRIER_USVIRGIN : GLLibConfig
- xplayer_CONN_TIMEOUT : GLLibConfig
- xplayer_ENABLE_DEBUG : GLLibConfig
- xplayer_ENABLE_DUAL_TCP : GLLibConfig
- xplayer_ENABLE_FIND_PLAYER : GLLibConfig
- xplayer_ENABLE_M7_SUPPORT : GLLibConfig
- xplayer_ENABLE_MULTIPLAYER : GLLibConfig
- xplayer_ENABLE_PLAYER_SPECIFIC_DATA : GLLibConfig
- xplayer_ENABLE_TIMEOUT : GLLibConfig
- xplayer_HTTP_NO_CANCEL : GLLibConfig
- xplayer_KEEP_ALIVE_TIME : GLLibConfig
- xplayer_USE_BUG_FIX_MESSAGE_SIZE : GLLibConfig
- xplayer_USE_HTTP_POST : GLLibConfig
- xplayer_XPLAYER_VERSION : GLLibConfig
- xRatio : ASprite
### - y -

- yRatio : **ASprite**