Offline Help Files

6.04.01 System Global Objects

The global object 'system' contains all the static functions related to system-wide functionality in Offline. This object has twelve associated methods, which are explained in this section.

TraceOutput

This method, which takes a message string as an argument, will trace a message to any of the following locations:

- Output window
- File
- Development debug window

The syntax is as follows:

```
System.TraceOutput(strMessage);
```

In the following example, we are displaying a message on a screen:

```
System.TraceOutput("This is a test.\n");
```

IsConnectionPackage

This method checks to see if the package received is a connection package. It takes no arguments and the syntax is as shown below:

```
System.IsConnectionPackage( );
```

This method can be used to trigger an action based on the result as shown in the following example:

```
if(System.IsConnectionPackage( ) == true) {
 // some action logic goes here:
}
```

IsSystemError

This method operates ina very similar manner to the IsConnectionPackage method previously introduced. It checks to see if there are any runtime errors, such as script parsing errors, execution errors or system function errors. The syntax is shown

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-10-05 22:50
URL: https://www.liquid-ui.com/knowledge_base/content/137/1082/en/60401-system-global-objects.html

below:

```
System.IsSystemError( );
```

In the following example, we will use this check to trigger a subsequent action based on the results of the check:

```
if(IsSystemError( ) {
// some action logic goes here:
 }
```

IsDeviceDocked

This method is used to determine if a handheld device is or is not docked to the system. The syntax is as shown below:

```
System.IsDeviceDocked( );
```

In the following example, we will use this method to check if the device is docked and then trigger subsequent actions based on the results returned.

```
if(System.IsDeviceDocked( ) == true) {
  // some action logic goes here:
 }
```

Encrypt

The Encrypt string method is used to return an encrypted string. It takes the string in unencrypted form as an argument and the syntax is as follows:

```
System.Encrypt(strPlain);
```

In the following example we will encrypt a plain-text string:

```
strEncrypt = System.Encrypt("unencrypted string");
```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-10-05 22:50
URL: https://www.liquid-ui.com/knowledge_base/content/137/1082/en/60401-system-global-objects.html

Decrypt

The string Decrypt method is the reverse of the Encrypt method. This is used to decrypt an encrypted string, taking the encrypted string as the argument. The syntax is as follows:

```
System.Decrypt(strEncrypt);
```

In the following example, we will pass an encrypted string, decrypt it and store the result in a variable:

```
strPlain = System.Decrypt("encrypted string");
```

LoadFile

This Boolean method will load and run a Javascript file. The method will return regardless of whether the script succeeds or fails. The filename is the argument and the syntax is as follows:

```
System.LoadFile(strFileName);
```

In the following example we will run a JavaScript file and base the resulting action on whether or not the file can load and run:

```
if(System.LoadFile("filename.js") == true)
{
 // Some action logic goes here:
}
else {
 // Some error handling logic goes here :
}
```

ParseGuiXTScript

Similar to the LoadFile method above, this method will parse in a JavaScript file and run it. Like the LoadFile method, it will return whether teh script succeeds or fails. This method takes the fielname as the argument in an array and the syntax is as follows:

```
System.ParseGuiXTScript(strGuiXTScriptFileName);
```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-10-05 22:50
URL: https://www.liquid-ui.com/knowledge_base/content/137/1082/en/60401-system-global-objects.html

This is demonstrated as shown below:

```
arScreenDefinition = arScreenDefinition.concat([System.ParseGuiXTScrip
t(strGuiXTScriptFileN ame)]);
```

SetActiveTheme

This method sets the active therme for SAP GUI, taking the theme name as a string argument. It is recommended to set the theme to 'System Dependent'. THe syntax is as shown below:

```
System.SetActiveTheme(strTheme);
```

In the following example, we will set the theme to 'High Contrast':

```
System.SetActiveTheme("HIGH CONTRAST THEME");
```

IsClientSupport

This Boolean method is used to check if a particular client supports certain features. This is achieved by checking the support bit from the client, which is taken as an argument. The syntax is shown below:

```
System.IsClientSupport(iSupportBit);
```

In the following example, we will pass the support bit 0x01 and check the system support:

```
System.IsSystemSupport(0x01);
```

StartCheckingConnection

This method starts a new thread to check the conenction with the server, taking the server name as an argument. The syntax is as follows:

```
System.StartCheckingConnection(strServerName);
```


(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-10-05 22:50
URL: https://www.liquid-ui.com/knowledge_base/content/137/1082/en/60401-system-global-objects.html

An example is shown below:

```
System.StartCheckingConnection("server name");
```

GetServerConnectionStatus

This method gets the connection status between the client and the server. The syntax is as follows:

```
System.GetServerCOnnectionStatus( );
```

In the following example, we will check the connection status and if true, perform additional tasks.

```
if(System.GetServerConnectionStatus( ) == true) {
// Additional task logic goes here:
}
```

Unique solution ID: #2085
Author: sarvani.kusuri@guixt.com
Last update: 2021-06-03 18:30

(c) 2026 Liquid UI | Synactive | GuiXT <dev@guixt.com> | 2026-10-05 22:50
URL: https://www.liquid-ui.com/knowledge_base/content/137/1082/en/60401-system-global-objects.html