

roHttpServer

ON THIS PAGE

- [ifHttpServer](#)
 - [GetFailureReason\(\) As String](#)
 - [AddGetFromString\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddGetFromFile\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddGetFromFolder\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddGetFromEvent\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddPostToString\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddPostToFile\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddPostToFormData\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddMethodFromEvent\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddMethodToFile\(parameters As roAssociativeArray\) As Boolean](#)
 - [AddMethodToString\(parameters As roAssociativeArray\) As Boolean](#)
 - [SetupDWSLink\(title As String\) As Boolean](#)
- [Parameters for "Add" Handler Methods](#)
- [ifMessagePort](#)
 - [SetPort\(port As roMessagePort\)](#)

OS8

- [OS8](#)
- [Version 7.1](#)
- [Version 7.0](#)
- [Version 6.2](#)
- [Version 6.1](#)
- [Previous Versions](#)

This object allows for processing of RESTful HTTP/HTTPS requests from remote URLs to the embedded web server of the BrightSign player. Many of the requests are provided to the script as *roHttpEvent* objects for handling.

Object Creation: The *roHttpServer* object is created with an *roAssociativeArray*.

```
CreateObject("roHttpServer", parameters As roAssociativeArray)
```

The associative array can contain the following parameters:

- [int] port: The port number of the embedded web server
- [roAssociativeArray] http: A set of HTTPS configuration parameters:
 - [string] key_file: The name of a file containing the private key in PEM format. If neither the `passphrase` nor `obfuscated_passphrase` are passed, then the key file must be unencrypted.
 - [string] certificate_file: The name of a file containing the certificate in PEM format
 - [string] passphrase: The passphrase to decrypt the key file
 - [string] obfuscated_passphrase: The obfuscated passphrase to decrypt the key file. Contact support@brightsign.biz to learn more about generating a key for obfuscation and storing it on the player.

Note

If the key and certificate are contained in a single file, it may be passed as the `certificate_file` (and the `key_file` can be omitted).

Example

```
server = CreateObject("roHttpServer", { port: 443, http: {  
certificate_file: "cert.pem", passphrase: "sekrit" } })
```

ifHttpServer

Each "Add" handler method described below takes an associative array as its parameter. Values in the associative array specify how the handler behaves. See the [table](#) at the end of this section for common `key:value` pairs.

GetFailureReason() As String

Yields additional useful information if an *roHttpServer* method fails.

AddGetFromString(parameters As roAssociativeArray) As Boolean

Causes any HTTP GET requests for the specified URL path to be met directly with the contents of the "body" member of the parameter associative array. The MIME type (and potentially the entire character set) should be specified if the request is expected to come from a web browser. The request is handled entirely within the *roHttpServer* method; no events are sent to the message port.

AddGetFromFile(parameters As roAssociativeArray) As Boolean

Causes any HTTP GET requests for the specified URL path to be met directly from the specified file. You should always specify the MIME type (and possibly the character set) if you expect the request to come from a web browser. The request is handled entirely within the *roHttpServer* method; no events are sent to the message port.

AddGetFromFolder(parameters As roAssociativeArray) As Boolean

Constructs a dynamic handler that serves up static files, which will appear as children of the defined storage folder. This method accepts an associative array with the following parameters:

- `folder`: The file path of the folder that will act as the root directory of the server. If this parameter is absent, everything on the storage device will be served.
- `url_prefix`: The URL prefix under which files will be served. If this parameter is absent, the URL will match from root.
- `filters`: An array of filters. Each filter is an associative array with the following parameters:
 - `re`: The regular expression to match against the request URL.
 - `ext`: The extension to match against the leaf file of the request URL.

If the "re" and "ext" parameters are absent, the filter will match everything.

- `headers`: An associative array containing arbitrary headers to be included with the automatic response.
- `content_type`: The contents of the "Content-Type" header included with the automatic response. This cannot be set in the same filter as the headers. The MIME type and character set can be specified together (e.g. "text/plain; charset=utf-8").

AddGetFromEvent(parameters As roAssociativeArray) As Boolean

Requests that an event of type *roHttpEvent* be sent to the configured message port. This occurs when an HTTP GET request is made for the specified URL path.

AddPostToString(parameters As roAssociativeArray) As Boolean

Requests that an event of type *roHttpEvent* be sent to the configured message port. This occurs when an HTTP POST request is made for the specified URL path. Use the *roHttpEvent.GetRequestBodyString()* method to retrieve the posted body.

AddPostToFile(parameters As roAssociativeArray) As Boolean

Requests that, when an HTTP POST request is made to the specified URL path, the request body be stored in a temporary file according to the `parameters["destination_directory"]` value in the associative array. When this request is complete, an `roHttpRequestEvent` event is sent to the configured message port. Use the `roHttpRequestEvent.GetRequestBodyFile()` method to retrieve the name of the temporary file. If the file still exists at the time the response is sent, it will be automatically deleted. However, if the player reboots or loses power during the POST process, the file will not be deleted. For this reason, we recommend using a dedicated subdirectory as the "destination_directory" and wiping this subdirectory during startup (using `DeleteDirectory()`) before adding handlers that refer to it.

AddPostToFormData(parameters As roAssociativeArray) As Boolean

Requests that, when an HTTP POST request is made to the specified URL path, an attempt be made to store form data (passed as `application/x-www-form-urlencoded` or `multipart/form-data`) in an associative array that can be retrieved by calling the `roHttpRequestEvent.GetFormData()` method.

AddMethodFromEvent(parameters As roAssociativeArray) As Boolean

Requests that an event of type `roHttpRequestEvent` be sent to the configured message port. Unlike `AddGetFromEvent()`, this method can support arbitrary HTTP methods. The HTTP method is specified using the `method` member in the associative array.

AddMethodToFile(parameters As roAssociativeArray) As Boolean

Requests that, when an arbitrary HTTP request is made to the specified URL path, the request body be stored in a temporary file according to the `parameters["destination_directory"]` value in the associative array. The HTTP method is specified using the `method` member in the associative array. When the request is complete, an `roHttpRequestEvent` event is sent to the configured message port. Use the `roHttpRequestEvent.GetRequestBodyFile()` method to retrieve the name of the temporary file. If the file still exists at the time the response is sent, it will be automatically deleted.

AddMethodToString(parameters As roAssociativeArray) As Boolean

Attempts to support an arbitrary HTTP method. The request body is placed in a string and an event is raised. This makes the request body available via the `roHttpRequestEvent.GetRequestBodyString()` method. A response can be sent in the same manner as the `AddGetToEvent()` method.

SetupDWSLink(title As String) As Boolean

Generates a tab in the [Diagnostic Web Server](#) (DWS) that links directly to the base `<ip_address:port>` of the `roHttpRequestServer` instance. The passed string specifies the title of the tab.

Example

```
server1.SetupDWSLink("My AWS Link")
server2.SetupDWSLink("My Other AWS Link")
```

PARAMETERS FOR "ADD" HANDLER METHODS

The following table describes common key:value pairs for "Add" handler methods:

Name	Applies to	Value
<code>url_path</code>	All	The path for which the handler method will be used
<code>user_data</code>	<code>GetFromEvent()</code> <code>PostToString()</code> <code>PostToFile()</code> <code>MethodToString()</code>	A user-defined value that can be retrieved by calling <code>roHttpRequestEvent.GetUserData()</code>
<code>method</code>	<code>AddMethodFromEvent()</code> <code>AddMethodToFile()</code>	The HTTP method associated with the generated <code>roHttpRequestEvent</code> . The method type can then be retrieved using <code>roHttpRequestEvent.GetMethod()</code> .
<code>passwords</code>	All	An associative array that contains a mapping between usernames and passwords

auth	All	The authentication type to use when passwords are set. This value can be either "basic" or "digest". The value defaults to "digest" if not specified.
realm	All	The authentication realm, which will be displayed by web browsers when prompting for a username and password
headers	GetFromFile()	An associative array that contains arbitrary headers to be included with the automated response
content_type	GetFromFile()	The contents of the "Content-Type" header that is included with the automated response. This may not be set at the same time as the headers member. You can set both the MIME type and character set together (e.g. "text/plain; charset=utf-8")
body	GetFromString()	The response body
filename	GetFromFile()	The path to the file used for the response body.
destination_directory	PostToFile()	The path to the directory used for the temporary file containing the request body. A random filename will be generated automatically.

ifMessagePort

SetPort(port As roMessagePort)

Posts messages of type *roHttpEvent* to the attached message port.