

# Brook framework

Silvio Clecio, Luciano Souza

May 14, 2014

# Contents

<b>1</b>	<b>Unit BrookAction</b>	<b>4</b>
1.1	Description . . . . .	4
1.2	Uses . . . . .	4
1.3	Overview . . . . .	4
1.4	Classes, Interfaces, Objects and Records . . . . .	5
1.5	Types . . . . .	13
<b>2</b>	<b>Unit BrookApplication</b>	<b>14</b>
2.1	Description . . . . .	14
2.2	Uses . . . . .	14
2.3	Overview . . . . .	14
2.4	Classes, Interfaces, Objects and Records . . . . .	15
2.5	Functions and Procedures . . . . .	16
<b>3</b>	<b>Unit BrookClasses</b>	<b>17</b>
3.1	Description . . . . .	17
3.2	Uses . . . . .	17
3.3	Overview . . . . .	17
3.4	Classes, Interfaces, Objects and Records . . . . .	17
3.5	Types . . . . .	18
<b>4</b>	<b>Unit BrookConfigurator</b>	<b>19</b>
4.1	Description . . . . .	19
4.2	Uses . . . . .	19
4.3	Overview . . . . .	19
4.4	Classes, Interfaces, Objects and Records . . . . .	20
4.5	Types . . . . .	22
<b>5</b>	<b>Unit BrookConstraints</b>	<b>23</b>
5.1	Description . . . . .	23
5.2	Uses . . . . .	23
5.3	Overview . . . . .	23
5.4	Classes, Interfaces, Objects and Records . . . . .	24
5.5	Types . . . . .	28

<b>6</b>	<b>Unit BrookConsts</b>	<b>29</b>
6.1	Description . . . . .	29
6.2	Constants . . . . .	29
6.3	Variables . . . . .	49
<b>7</b>	<b>Unit BrookException</b>	<b>51</b>
7.1	Description . . . . .	51
7.2	Uses . . . . .	51
7.3	Overview . . . . .	51
7.4	Classes, Interfaces, Objects and Records . . . . .	51
7.5	Types . . . . .	53
<b>8</b>	<b>Unit BrookHttpClient</b>	<b>54</b>
8.1	Description . . . . .	54
8.2	Uses . . . . .	54
8.3	Overview . . . . .	54
8.4	Classes, Interfaces, Objects and Records . . . . .	55
8.5	Types . . . . .	62
<b>9</b>	<b>Unit BrookHttpConsts</b>	<b>63</b>
9.1	Description . . . . .	63
9.2	Uses . . . . .	63
9.3	Constants . . . . .	63
9.4	Variables . . . . .	80
<b>10</b>	<b>Unit BrookHttpDefs</b>	<b>81</b>
10.1	Description . . . . .	81
10.2	Uses . . . . .	81
10.3	Types . . . . .	81
<b>11</b>	<b>Unit BrookHttpUtils</b>	<b>83</b>
11.1	Description . . . . .	83
11.2	Uses . . . . .	83
11.3	Overview . . . . .	83
11.4	Functions and Procedures . . . . .	84
11.5	Types . . . . .	86
<b>12</b>	<b>Unit BrookMessages</b>	<b>87</b>
12.1	Description . . . . .	87
12.2	Variables . . . . .	87
<b>13</b>	<b>Unit BrookMiddleware</b>	<b>90</b>
13.1	Description . . . . .	90
13.2	Uses . . . . .	90
13.3	Overview . . . . .	90
13.4	Classes, Interfaces, Objects and Records . . . . .	91
13.5	Types . . . . .	92

<b>14 Unit BrookRouter</b>	<b>93</b>
14.1 Description . . . . .	93
14.2 Uses . . . . .	93
14.3 Overview . . . . .	94
14.4 Classes, Interfaces, Objects and Records . . . . .	94
14.5 Types . . . . .	99
<b>15 Unit BrookSession</b>	<b>101</b>
15.1 Description . . . . .	101
15.2 Uses . . . . .	101
15.3 Overview . . . . .	101
15.4 Classes, Interfaces, Objects and Records . . . . .	102
15.5 Types . . . . .	105
<b>16 Unit BrookUtils</b>	<b>106</b>
16.1 Description . . . . .	106
16.2 Uses . . . . .	106
16.3 Overview . . . . .	106
16.4 Classes, Interfaces, Objects and Records . . . . .	108
16.5 Functions and Procedures . . . . .	110
16.6 Types . . . . .	116
16.7 Variables . . . . .	117

# Chapter 1

## Unit BrookAction

### 1.1 Description

Brook framework, Action Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 1.2 Uses

- BrookClasses(3)
- BrookHttpDefs(10)
- BrookException(7)
- BrookMessages(12)
- BrookUtils(16)
- BrookHTTPUtils(11)
- BrookHTTPConsts(9)
- Classes
- SysUtils

### 1.3 Overview

`EBrookAction` Class Handles exceptions for `TBrookAction`(1.4).

`TBrookAction` Class Provides features to handle HTTP requests and responses.

**TBrookGAction Class** Provides features to handle HTTP requests and responses mapping URIs to object.

## 1.4 Classes, Interfaces, Objects and Records

### EBrookAction Class

---

#### Hierarchy

EBrookAction > EBrook(7.4) > Exception

#### Description

Handles exceptions for TBrookAction(1.4).

### TBrookAction Class

---

#### Hierarchy

TBrookAction > TBrookObject(3.4) > TObject

#### Description

Provides features to handle HTTP requests and responses.

#### Properties

**Files** public property Files: TBrookUploadedFiles read FFiles;  
The list of files coming from a request called by the POST method.

**Fields** public property Fields: TStrings read FFields;  
The list of variables coming from a request called by the POST method.

**Params** public property Params: TStrings read FParams;  
The list of variables coming from a request called by the GET method.

**Values** public property Values: TStrings read FValues;  
The list of variables received from a parametrized URL.

**Method** public property Method: string read GetMethod;  
Returns the HTTP request method.

#### Methods

##### Create

**Declaration** public constructor Create; overload; virtual;

**Description** Creates an instance of a TBrookAction(1.4) class.

## Create

**Declaration** `public constructor Create(ARequest: TBrookRequest; AResponse: TBrookResponse); overload; virtual;`

**Description** Creates an instance of a TBrookAction(1.4) class passing params to request/response.

## Destroy

**Declaration** `public destructor Destroy; override;`

**Description** Frees an instance of TBrookAction(1.4) class.

## Register

**Declaration** `public class procedure Register(const APattern: string; const ADefault: Boolean = False); overload;`

**Description** Registers an action.

**Parameters** **APattern** Is an expression defining which URLs is used to call an action. It is possible to use variables inside URLs:

**:name** – Represents a variable that spans single URL component between slashes.

Examples:

```
TMyAction.Register('/foo/:myvar');
```

Value of a variable "myvar" can be read from the property Values(1.4), e.g.:

```
Write(Values.Values['myvar']);
```

Any number of variables can be combined:

```
TMyAction.Register('/foo/:cat/:id');
```

**\*name** – Represents a variable that spans one or more levels between slashes in the current

Examples:

```
TMyAction.Register('/home/*path');
```

Any of the following URLs will match:

```
http://localhost/cgi-bin/cgi1/home/file
```

```
http://localhost/cgi-bin/cgi1/home/dir/file
```

```
http://localhost/cgi-bin/cgi1/home/dir/subdir/file etc.
```

Variable Values.Values['path'] will receive 'file', 'dir/file' or 'dir/subdir/file' correspondingly.

You can also add static text after variable part:

```
TMyAction.Register('/home/*path/download');
```

```
http://localhost/cgi-bin/cgi1/home/dir/file/download – This will match,
```

```
http://localhost/cgi-bin/cgi1/home/dir/file/info – but not this, because ending is different.
```

Multi-level variable can be combined with any number of single-level variables in any order:

```
TMyAction.Register('/home/user/:uid/file/*fpath/version/:vid/info');
```

**NOTE:** Only one multi-level variable can be specified per URL.

**url/** – Adds a slash to the end of the URL if does not exist. Example:

```
TMyAction.Register('/foo/');
```

An action can be accessed as `http://localhost/cgi-bin/cgi1/foo` or `http://localhost/cgi-bin/cgi1/foo/`. When called as `http://localhost/cgi-bin/cgi1/foo`, it will be automatically redirected to `http://localhost/cgi-bin/cgi1/foo/`. If the pathinfo is different from `/foo` a 404 page is returned;

**NOTE:** Two actions can't be registered with the same pattern except when they are called by means of different HTTP methods.

**ADefault** A action registered as *Default* will be called automatically if the URL does not match with *Pattern* of any registered actions. It is not allowed to register more than one action as default. A typical example of use is:

```
TMyAction.Register('*', True);
```

## Register

**Declaration** `public class procedure Register(const APattern: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean = False); overload;`

**Description** Registers an action specifying the HTTP request method.

**Parameters** **AMethod** Informs the HTTP request method being valid the following options: `rmAll`, `rmGet`, `rmHead`, `rmOptions`, `rmPost`, `rmPut` or `rmDelete`. The only way to register two actions with the same pattern is differentiating the value of this parameter. If at least one action has this parameter changed, the route mapping is enabled in `Mapped(16.4)`. A typical example of use is:

```
procedure TMyAction1.Get;
begin
  Write('GET');
end;
```

```
procedure TMyAction1.Put;
begin
  Write('PUT');
end;
```

```
procedure TMyAction2.Post;
begin
  Write('POST');
end;
```

### initialization

```
TMyAction1.Register('/foo1', rmGet);
TMyAction1.Register('/foo1', rmPut);
TMyAction2.Register('/foo1', rmPost)
```



## GetPath

**Declaration** `public class function GetPath: string;`

**Description** Returns the path of action. Exemple:  
`/cgi-bin/cgi1/myaction.`

## DoRequest

**Declaration** `public procedure DoRequest(ARequest: TBrookRequest; AResponse: TBrookResponse); virtual;`

**Description** Calls the method `Request(1.4)`.

## Request

**Declaration** `public procedure Request(ARequest: TBrookRequest; AResponse: TBrookResponse); virtual;`

**Description** Is triggered by a request of any HTTP method.

## GetFields

**Declaration** `public procedure GetFields(AObject: TObject);`

**Description** Get an object with the fields coming from a `x-www-form-urlencoded` form.

## GetParams

**Declaration** `public procedure GetParams(AObject: TObject);`

**Description** Get an object with the params coming from a `QUERY_STRING`.

## GetValues

**Declaration** `public procedure GetValues(AObject: TObject);`

**Description** Get an object with the variables coming from an URL.

## UrlFor

**Declaration** `public function UrlFor(AActionClass: TBrookActionClass): string; overload;`

**Description** Creates an URL for action.

## UrlFor

**Declaration** public function UrlFor(AActionClass: TBrookActionClass; const AParams: array of string): string; overload;

**Description** Creates an URL for an action informing an array of parameters. Exemple:

```
procedure TMyAction.Get;
begin
  // When calling with http://localhost/cgi-bin/cgi1/foo/myvalue
  // the output will be /cgi-bin/cgi1/foo/myvalue
  Write(UrlFor(TMyAction, ['myvalue']));
end;

initialization
  TMyAction.Register('/foo/:myvar')
```

## UrlFor

**Declaration** public function UrlFor(AClassName: string; const AParams: array of string): string; overload;

**Description** Creates an URL for an action passing an array of parameters however informing the class name as string.

## UrlFor

**Declaration** public function UrlFor(AClassName: string): string; overload;

**Description** Creates an URL for an action informing the class name as string.

## Get

**Declaration** public procedure Get; virtual;

**Description** Is triggered by a GET HTTP request method.

## Post

**Declaration** public procedure Post; virtual;

**Description** Is triggered by a POST HTTP request method.

## Put

**Declaration** public procedure Put; virtual;

**Description** Is triggered by a PUT HTTP request method.

## Delete

**Declaration** `public procedure Delete; virtual;`

**Description** Is triggered by a DELETE HTTP request method.

## Head

**Declaration** `public procedure Head; virtual;`

**Description** Is triggered by a HEAD HTTP request method.

## Options

**Declaration** `public procedure Options; virtual;`

**Description** Is triggered by an OPTIONS HTTP request method.

## Redirect

**Declaration** `public procedure Redirect(const AUrl: string); overload;`

**Description** Redirects to an URL.

## Redirect

**Declaration** `public procedure Redirect(const AUrl: string; const AStatusCode: Word);  
overload;`

**Description** Redirects to an URL informing the (302, 307) status code.

## Redirect

**Declaration** `public procedure Redirect(const AUrl: string; const AUseRootUrl: Boolean);  
overload;`

**Description** Redirects to an URL informing the root URL.

## Redirect

**Declaration** `public procedure Redirect(const AUrl: string; const AUseRootUrl: Boolean;  
const AStatusCode: Word); overload;`

**Description** Redirects to an URL informing the (302, 307) status code and the ScriptName.

## Error

**Declaration** `public procedure Error(const AMsg: string); overload;`

**Description** Raises a message for action exceptions.

## Error

**Declaration** `public procedure Error(const AMsg: string; const AArgs: array of const); overload;`

**Description** Raises a formatted message for action exceptions.

## Stop

**Declaration** `public procedure Stop(const AMsg: string); overload;`

**Description** Stops the action showing an exception message.

## Stop

**Declaration** `public procedure Stop(const AMsg: string; const AArgs: array of const); overload;`

**Description** Stops the action showing a formatted exception message.

## Render

**Declaration** `public procedure Render(const AFileName: TFileName); overload; virtual;`

**Description** Writes the content of a file.

## Render

**Declaration** `public procedure Render(const AFileName: TFileName; const AArgs: array of const); overload; virtual;`

**Description** Writes the content of a file passing parameters to the output.

## Clear

**Declaration** `public procedure Clear;`

**Description** Clears all written content with `Write()`, `WriteLn()`, `Render()` etc.

## Exists

**Declaration** `public function Exists(const AName: string): Boolean;`

**Description** Checks if a name exists in fields.

## Write

**Declaration** `public procedure Write(const AString: string); overload;`

**Description** Writes a string.

### Write

**Declaration** `public procedure Write(const ABoolean: Boolean); overload;`

**Description** Writes a boolean.

### Write

**Declaration** `public procedure Write(const AInteger: Integer); overload;`

**Description** Writes an integer.

### Write

**Declaration** `public procedure Write(const AFloat: Double); overload;`

**Description** Writes a float.

### Write

**Declaration** `public procedure Write(AObject: TObject); overload;`

**Description** Writes an object.

### Write

**Declaration** `public procedure Write(AStream: TStream); overload;`

**Description** Writes a content of stream.

### Write

**Declaration** `public procedure Write(const AFmt: string; const AArgs: array of const);  
overload;`

**Description** Writes a formatted string.

## TBrookGAction Class ---

### Hierarchy

TBrookGAction > TBrookAction(1.4) > TBrookObject(3.4) > TObject

### Description

Provides features to handle HTTP requests and responses mapping URIs to object.

## Properties

**Entity** public property Entity: T read FEntity write FEntity;  
Maps URI to object.

## Methods

### Create

**Declaration** public constructor Create; overload; override;

**Description** Creates an instance of a TBrookGAction(1.4) class.

### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookGAction(1.4) class.

### Request

**Declaration** public procedure Request(ARequest: TBrookRequest; AResponse:  
TBrookResponse); override;

**Description** Is triggered by a request of any HTTP method.

## 1.5 Types

### TBrookActionClass

---

**Declaration** TBrookActionClass = class of TBrookAction;

**Description** Is a metaclass for TBrookAction(1.4) class.

## Chapter 2

# Unit BrookApplication

### 2.1 Description

Brook framework, Application Interface

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 2.2 Uses

- BrookConsts(6)
- BrookClasses(3)
- BrookException(7)
- BrookMessages(12)

### 2.3 Overview

`EBrookApplication` Class Handles exceptions for application.

`IBrookApplication` Interface Interface of the application.

`BrookApp` Returns the application instance.

`BrookRegisterApp` Register the application.

`BrookUnregisterApp` Unregister the application.

## 2.4 Classes, Interfaces, Objects and Records

### EBrookApplication Class

---

#### Hierarchy

EBrookApplication > EBrook(7.4) > Exception

#### Description

Handles exceptions for application.

### IBrookApplication Interface

---

#### Hierarchy

IBrookApplication > IBrookInterface(3.4) > IInterface

#### Description

Interface of the application.

#### Properties

**Terminated** `public property Terminated: Boolean read GetTerminated;`  
Checks whether the application is terminated.

#### Methods

##### GetTerminated

**Declaration** `public function GetTerminated: Boolean;`

**Description** Gets whether the application is terminated.

##### Instance

**Declaration** `public function Instance: TObject;`

**Description** Returns the instance of broker application.

##### Run

**Declaration** `public procedure Run;`

**Description** Initializes and runs the application.



## **Terminate**

**Declaration** `public procedure Terminate;`

**Description** Terminates the application.

## **2.5 Functions and Procedures**

### **BrookApp**

---

**Declaration** `function BrookApp: IBrookApplication;`

**Description** Returns the application instance.

### **BrookRegisterApp**

---

**Declaration** `procedure BrookRegisterApp(AApp: IBrookApplication);`

**Description** Register the application.

### **BrookUnregisterApp**

---

**Declaration** `procedure BrookUnregisterApp;`

**Description** Unregister the application.

# Chapter 3

## Unit BrookClasses

### 3.1 Description

Brook framework, Base Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 3.2 Uses

- BrookConsts(6)
- Classes

### 3.3 Overview

`IBrookInterface` Interface Is the main interface for Brook.

`TBrookObject` Class Is the main object for Brook.

`TBrookInterfacedObject` Class Is the main interfaced object for Brook.

`TBrookComponent` Class Is the main component for Brook.

### 3.4 Classes, Interfaces, Objects and Records

`IBrookInterface` Interface \_\_\_\_\_

**Hierarchy**

`IBrookInterface` > `IInterface`

## Description

Is the main interface for Brook.

## TBrookObject Class

---

### Hierarchy

TBrookObject > TObject

## Description

Is the main object for Brook.

## TBrookInterfacedObject Class

---

### Hierarchy

TBrookInterfacedObject > TInterfacedObject

## Description

Is the main interfaced object for Brook.

## TBrookComponent Class

---

### Hierarchy

TBrookComponent > TComponent

## Description

Is the main component for Brook.

## 3.5 Types

### TBrookClass

---

**Declaration** `TBrookClass = class of TBrookObject;`

**Description** Is the main class for Brook.

# Chapter 4

## Unit BrookConfigurator

### 4.1 Description

Brook framework, Configurator Class

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 4.2 Uses

- BrookClasses(3)
- BrookUtils(16)
- BrookException(7)
- BrookMessages(12)
- BrookConsts(6)
- Classes
- SysUtils

### 4.3 Overview

EBrookConfigurator Class Handles exceptions for TBrookConfigurator(4.4).

TBrookConfigurator Class Configures objects by means of string or file.

## 4.4 Classes, Interfaces, Objects and Records

### EBrookConfigurator Class

---

#### Hierarchy

EBrookConfigurator > EBrook(7.4) > Exception

#### Description

Handles exceptions for TBrookConfigurator(4.4).

### TBrookConfigurator Class

---

#### Hierarchy

TBrookConfigurator > TBrookObject(3.4) > TObject

#### Description

Configures objects by means of string or file.

#### Properties

**AutoLoaded**    public property AutoLoaded: Boolean read FAutoLoaded write FAutoLoaded;  
Defines if the configuration is read automatically.

**Configuration** public property Configuration: string read FConfiguration write FConfiguration;

Defines the configuration using one of the following formats. Exemple:

```
cfg.Configuration := 'param1=value1;param2=value2;param3=value2';
```

or:

```
cfg.Configuration := 'cgi1.cfg';
```

This option requires the following file format:

```
param1=value1
param2=value2
param3=value2
```

NOTE: If a configuration is passed to BrookSettings.Configuration, the data informed in this property will be disregarded.

**Target**        public property Target: TObject read FTarget write FTarget;  
Defines the object to be configured.

**Values**        public property Values[AName:string]: string read GetValues;  
Devolve o valor de um parmetro.

**ClassChecking** public property ClassChecking: Boolean read FClassChecking write FClassChecking;

Enables the class name checking, i.e., if the class of the configuring object is a 'TObject1', the following string will be informed as configuration:

'tobject1.param1=value1;tobject1.param2=value2;tobject1.param3=value3'

This feature is useful when more then one object will be configured in one or more configurator objects.

## Methods

### Create

**Declaration** public constructor Create;

**Description** Creates an instance of a TBrookConfigurator(4.4) class.

### Create

**Declaration** public constructor Create(const AConfiguration: string); reintroduce;

**Description** Creates an instance of a TBrookConfigurator(4.4) class passing the configuration in its parameter.

### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookConfigurator(4.4) class.

### Load

**Declaration** public function Load: TBrookConfigurator; virtual;

**Description** Loads the configuration file to the memory.

### Clear

**Declaration** public procedure Clear;

**Description** Clears the internal configuration.

### Configure

**Declaration** public procedure Configure;

**Description** Configures the object informed in the Target property.

## 4.5 Types

**TBrookConfiguratorClass** \_\_\_\_\_

**Declaration** `TBrookConfiguratorClass = class of TBrookConfigurator;`

**Description** Is a metaclass for TBrookConfigurator(4.4) class.

# Chapter 5

## Unit BrookConstraints

### 5.1 Description

Brook framework, Constraints Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 5.2 Uses

- BrookClasses(3)
- BrookAction(1)
- BrookRouter(14)
- BrookException(7)
- BrookMessages(12)
- Classes
- SysUtils

### 5.3 Overview

`EBrookConstraint` Class Handles exceptions for `TBrookConstraint`(5.4).

`TBrookCustomConstraint` Class Offers general features for constraint handling.

`TBrookConstraint` Class Offers features for constraint handling.

`TBrookConstraintItem` record Defines a constraint item.



`TBrookCustomConstraints` Class Registers constraint classes.

`TBrookConstraints` Class Registers and executes constraint classes.

## 5.4 Classes, Interfaces, Objects and Records

### EBrookConstraint Class

---

#### Hierarchy

`EBrookConstraint` > `EBrook(7.4)` > `Exception`

#### Description

Handles exceptions for `TBrookConstraint(5.4)`.

#### Properties

**Constraint** public property `Constraint`: `TBrookConstraint` read `FConstraint`;  
Offers an instance of the current constraint.

#### Methods

##### Create

**Declaration** public constructor `Create(AConstraint: TBrookConstraint; const AMsg: string); overload;`

**Description** Creates an instance of a `EBrookConstraint(5.4)` class.

##### CreateFmt

**Declaration** public constructor `CreateFmt(AConstraint: TBrookConstraint; const AMsg: string; const AArgs: array of const); overload;`

**Description** Creates an instance of `EBrookConstraint` with a formatted message.

##### Destroy

**Declaration** public destructor `Destroy; override;`

**Description** Frees an instance of `EBrookConstraint(5.4)` class.

### TBrookCustomConstraint Class

---

#### Hierarchy

`TBrookCustomConstraint` > `TBrookObject(3.4)` > `TObject`

## Description

Offers general features for constraint handling.

## Methods

### Error

**Declaration** `public procedure Error(const AMsg: string); overload;`

**Description** Raises a message for constraint exceptions.

### Error

**Declaration** `public procedure Error(const AMsg: string; const AArgs: array of const);  
overload;`

**Description** Raises a formatted message for constraint exceptions.

### Stop

**Declaration** `public procedure Stop(const AMsg: string); overload;`

**Description** Stops the code execution showing an exception message.

### Stop

**Declaration** `public procedure Stop(const AMsg: string; const AArgs: array of const);  
overload;`

**Description** Stops the code execution showing a formatted exception message.

### Execute

**Declaration** `public procedure Execute; virtual; abstract;`

**Description** Offers a abstract method for the user validations.

## TBrookConstraint Class ---

### Hierarchy

TBrookConstraint > TBrookCustomConstraint(5.4) > TBrookObject(3.4) > TObject

## Description

Offers features for constraint handling.

## Properties

**Action** public property Action: TBrookAction read FAction;  
Offers an instance of the current action.

**Route** public property Route: TBrookRoute read FRoute;  
Offers an instance of the current route.

## Methods

### Create

**Declaration** public constructor Create(AAction: TBrookAction; ARoute: TBrookRoute);  
virtual;

**Description** Creates an instance of a TBrookConstraint(5.4) class.

### Register

**Declaration** public class procedure Register(AActionClass: TBrookActionClass);

**Description** Register the constraint class.

## TBrookConstraintItem record ---

### Description

Defines a constraint item.

### Fields

**ActionClass** ActionClass: TBrookActionClass;

**ConstraintClass** ConstraintClass: TBrookConstraintClass;

## TBrookCustomConstraints Class ---

### Hierarchy

TBrookCustomConstraints > TBrookObject(3.4) > TObject

### Description

Registers constraint classes.

### Methods

#### Create

**Declaration** public constructor Create; virtual;

**Description** Creates an instance of a TBrookConstraints(5.4) class.

## Destroy

**Declaration** `public destructor Destroy; override;`

**Description** Frees an instance of `TBrookConstraints(5.4)` class.

## TBrookConstraints Class

---

### Hierarchy

`TBrookConstraints` > `TBrookCustomConstraints(5.4)` > `TBrookObject(3.4)` > `TObject`

### Description

Registers and executes constraint classes.

### Methods

#### RegisterService

**Declaration** `public class procedure RegisterService;`

**Description** Registers the service provided by this class.

#### UnregisterService

**Declaration** `public class procedure UnregisterService;`

**Description** Unregisters the service provided by this class.

#### Service

**Declaration** `public class function Service: TBrookConstraints;`

**Description** Return an instance of this class.

#### Add

**Declaration** `public procedure Add(AActionClass: TBrookActionClass; AConstraintClass: TBrookConstraintClass);`

**Description** Adds a constraint item.

#### Execute

**Declaration** `public procedure Execute(AAction: TBrookAction; ARoute: TBrookRoute); virtual;`

**Description** Triggers the user validations implemented in the constraint.

## 5.5 Types

### **TBrookCustomConstraintClass** \_\_\_\_\_

**Declaration** `TBrookCustomConstraintClass = class of TBrookCustomConstraint;`

**Description** Is a metaclass for TBrookCustomConstraint(5.4) class.

### **TBrookConstraintClass** \_\_\_\_\_

**Declaration** `TBrookConstraintClass = class of TBrookConstraint;`

**Description** Is a metaclass for TBrookConstraint(5.4) class.

### **TBrookCustomConstraintsClass** \_\_\_\_\_

**Declaration** `TBrookCustomConstraintsClass = class of TBrookCustomConstraints;`

**Description** Is a metaclass for TBrookCustomConstraints(5.4) class.

### **TBrookConstraintsClass** \_\_\_\_\_

**Declaration** `TBrookConstraintsClass = class of TBrookConstraints;`

**Description** Is a metaclass for TBrookConstraints(5.4) class.

### **PBrookConstraintItem** \_\_\_\_\_

**Declaration** `PBrookConstraintItem = ^TBrookConstraintItem;`

# Chapter 6

## Unit BrookConsts

### 6.1 Description

Brook framework, Consts Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 6.2 Constants

**ES** \_\_\_\_\_

**Declaration** ES = '';

**Description** Empty string.

**NS** \_\_\_\_\_

**Declaration** NS = 'null';

**Description** Null string.

**BR** \_\_\_\_\_

**Declaration** BR = '<br />';

**Description** BR HTML tag.

**HS** \_\_\_\_\_

**Declaration** HS = ': ';

**Description** Param separator.

## LE

---

**Declaration** LE = LineEnding;

**Description** Line ending.

## NU

---

**Declaration** NU = #0;

**Description** NULL

## HT

---

**Declaration** HT = #9;

**Description** TAB

## LF

---

**Declaration** LF = #10;

**Description** LF

## CR

---

**Declaration** CR = #13;

**Description** CR

## CRLF

---

**Declaration** CRLF = CR + LF;

**Description** CRLF

## NullDate

---

**Declaration** NullDate = 0;

**Description** Null date.

## NullTime

---

**Declaration** NullTime = 0;

**Description** Null time.

## NullDateTime

---

**Declaration** NullDateTime = 0;

**Description** Null datetime.

## SP

---

**Declaration** SP = #32;

**Description** Space.

## DQ

---

**Declaration** DQ = #34;

**Description** ”

## PO

---

**Declaration** PO = #35;

**Description** #

## PT

---

**Declaration** PT = #37;

**Description** %

## AM

---

**Declaration** AM = #38;

**Description** &

## AP

---

**Declaration** AP = #39;

**Description** ’

## PS

---

**Declaration** PS = #40;

**Description** (



**PE** \_\_\_\_\_

**Declaration** PE = #41;

**Description** )

**AK** \_\_\_\_\_

**Declaration** AK = #42;

**Description** \*

**CS** \_\_\_\_\_

**Declaration** CS = #44;

**Description** ,

**DT** \_\_\_\_\_

**Declaration** DT = #46;

**Description** .

**US** \_\_\_\_\_

**Declaration** US = #47;

**Description** /

**CO** \_\_\_\_\_

**Declaration** CO = #58;

**Description** :

**SC** \_\_\_\_\_

**Declaration** SC = #59;

**Description** ;

**LT** \_\_\_\_\_

**Declaration** LT = #60;

**Description** <

## EQ

---

Declaration EQ = #61;

Description =

## GT

---

Declaration GT = #62;

Description >

## QU

---

Declaration QU = #63;

Description ?

## AT

---

Declaration AT = #64;

Description @

## DS

---

Declaration DS = #92;

Description \

## LS

---

Declaration LS = #91;

Description [

## RS

---

Declaration RS = #93;

Description ]

## BROOK\_GUID

---

Declaration BROOK\_GUID = '{D2BDD8EF-78C0-47CD-95C3-664CDFDAA9E}';

## BROOK\_APP\_GUID

---

Declaration BROOK\_APP\_GUID = '{669B03B7-AA2D-4B64-AAFC-4FBD4A41267E}';

## **BROOK\_ERROR\_MASK**

---

**Declaration** BROOK\_ERROR\_MASK = '%s: %s';

## **BROOK\_GMT\_FRMT**

---

**Declaration** BROOK\_GMT\_FRMT = '%s, %.2d %s %d %.2d:%.2d:%.2d GMT';

## **BROOK\_UUID\_MASK**

---

**Declaration** BROOK\_UUID\_MASK = '%.8x%.4x%.4x%.2x%.2x%.2x%.2x%.2x%.2x%.2x';

## **BROOK\_UUID\_SEP\_MASK**

---

**Declaration** BROOK\_UUID\_SEP\_MASK = '%.8x-%.4x-%.4x-%.2x%.2x%.2x%.2x%.2x%.2x';

## **BROOK\_OVERRIDDEN\_METHOD**

---

**Declaration** BROOK\_OVERRIDDEN\_METHOD = 'METHOD';

## **BROOK\_SESS\_ID**

---

**Declaration** BROOK\_SESS\_ID = 'BRKSESSID';

## **BROOK\_SESS\_PREFIX**

---

**Declaration** BROOK\_SESS\_PREFIX = 'brook\_sess\_';

## **BROOK\_SESS\_DEFAULT\_TIMEOUT**

---

**Declaration** BROOK\_SESS\_DEFAULT\_TIMEOUT = 3 \* 60;

**Description** 3 minutes.

## **BROOK\_MAX\_MIME\_TYPE**

---

**Declaration** BROOK\_MAX\_MIME\_TYPE = 682;

## **BROOK\_MIME\_TYPE**

---

**Declaration** BROOK\_MIME\_TYPE: array[0..BROOK\_MAX\_MIME\_TYPE, 0..2] of string = ( ('3D Crossword Plugin', 'application/vnd.hzn-3d-crossword', '.x3d'), ('3GP', 'video/3gpp', '.3gp'), ('3GP2', 'video/3gpp2', '.3g2'), ('3GPP MSEQ File', 'application/vnd.mseq', '.mseq'), ('3M Post It Notes', 'application/vnd.3m.post-it-notes', '.pwn'), ('3rd Generation Partnership Project - Pic Large', 'application/vnd.3gpp.pic-bw-large', '.plb'), ('3rd Generation Partnership Project - Pic Small', 'application/vnd.3gpp.pic-bw-small', '.psb'), ('3rd Generation Partnership Project - Pic Var', 'application/vnd.3gpp.pic-bw-var', '.pvb'), ('3rd

Generation Partnership Project - Transaction Capabilities Application Part',  
 'application/vnd.3gpp2.tcap', '.tcap'), ('7-Zip',  
 'application/x-7z-compressed', '.7z'), ('AbiWord', 'application/x-abiword',  
 '.abw'), ('Ace Archive', 'application/x-ace-compressed', '.ace'), ('Active  
 Content Compression', 'application/vnd.americandynamics.acc', '.acc'), ('ACU  
 Cobol', 'application/vnd.acucobol', '.acu'), ('ACU Cobol',  
 'application/vnd.acucorp', '.atc'), ('Adaptive differential pulse-code  
 modulation', 'audio/adpcm', '.adp'), ('Adobe (Macropedia) Authorware - Binary  
 File', 'application/x-authorware-bin', '.aab'), ('Adobe (Macropedia)  
 Authorware - Map', 'application/x-authorware-map', '.aam'), ('Adobe  
 (Macropedia) Authorware - Segment File', 'application/x-authorware-seg',  
 '.aas'), ('Adobe AIR Application',  
 'application/vnd.adobe.air-application-installer-package+zip', '.air'),  
 ('Adobe Flash', 'application/x-shockwave-flash', '.swf'), ('Adobe Flex  
 Project', 'application/vnd.adobe.fxp', '.fxp'), ('Adobe Portable Document  
 Format', 'application/pdf', '.pdf'), ('Adobe PostScript Printer Description  
 File Format', 'application/vnd.cups-ppd', '.ppd'), ('Adobe Shockwave Player',  
 'application/x-director', '.dir'), ('Adobe XML Data Package',  
 'application/vnd.adobe.xdp+xml', '.xdp'), ('Adobe XML Forms Data Format',  
 'application/vnd.adobe.xfdf', '.xfdf'), ('Advanced Audio Coding (AAC)',  
 'audio/x-aac', '.aac'), ('Ahead AIR Application',  
 'application/vnd.ahead.space', '.ahead'), ('AirZip FileSECURE',  
 'application/vnd.airzip.filesecure.azf', '.azf'), ('AirZip FileSECURE',  
 'application/vnd.airzip.filesecure.azs', '.azs'), ('Amazon Kindle eBook  
 format', 'application/vnd.amazon.ebook', '.azw'), ('AmigaDE',  
 'application/vnd.amiga.ami', '.ami'), ('Andrew Toolkit',  
 'application/andrew-inset', 'N/A'), ('Android Package Archive',  
 'application/vnd.android.package-archive', '.apk'), ('ANSER-WEB Terminal  
 Client - Certificate Issue',  
 'application/vnd.anser-web-certificate-issue-initiation', '.cii'),  
 ('ANSER-WEB Terminal Client - Web Funds Transfer',  
 'application/vnd.anser-web-funds-transfer-initiation', '.fti'), ('Antix Game  
 Player', 'application/vnd.antix.game-component', '.atx'), ('Apple Installer  
 Package', 'application/vnd.apple.installer+xml', '.mpkg'), ('Applixware',  
 'application/applixware', '.aw'), ('Archipelago Lesson Player',  
 'application/vnd.hhe.lesson-player', '.les'), ('Arista Networks Software  
 Image', 'application/vnd.aristanetworks.swi', '.swi'), ('Assembler Source  
 File', 'text/x-asm', '.s'), ('Atom Publishing Protocol',  
 'application/atomcat+xml', '.atomcat'), ('Atom Publishing Protocol Service  
 Document', 'application/atomsvc+xml', '.atomsvc'), ('Atom Syndication  
 Format', 'application/atom+xml', '.atom.xml'), ('Attribute Certificate',  
 'application/pkix-attr-cert', '.ac'), ('Audio Interchange File Format',  
 'audio/x-aiff', '.aif'), ('Audio Video Interleave (AVI)', 'video/x-msvideo',  
 '.avi'), ('Audiograph', 'application/vnd.audiograph', '.aep'), ('AutoCAD  
 DXF', 'image/vnd.dxf', '.dxf'), ('Autodesk Design Web Format (DWF)',  
 'model/vnd.dwf', '.dwf'), ('Binary CPIO Archive', 'application/x-bcpio',

'.bcpio'), ('Binary Data', 'application/octet-stream', '.bin'), ('Bitmap Image File', 'image/bmp', '.bmp'), ('BitTorrent', 'application/x-bittorrent', '.torrent'), ('Blackberry COD File', 'application/vnd.rim.cod', '.cod'), ('Blueice Research Multipass', 'application/vnd.blueice.multipass', '.mpm'), ('BMI Drawing Data Interchange', 'application/vnd.bmi', '.bmi'), ('Bourne Shell Script', 'application/x-sh', '.sh'), ('BTIF', 'image/prs.btif', '.btif'), ('BusinessObjects', 'application/vnd.businessobjects', '.rep'), ('Bzip Archive', 'application/x-bzip', '.bz'), ('Bzip2 Archive', 'application/x-bzip2', '.bz2'), ('C Shell Script', 'application/x-csh', '.csh'), ('C Source File', 'text/x-c', '.c'), ('CambridgeSoft Chem Draw', 'application/vnd.chemdraw+xml', '.cdxml'), ('Cascading Style Sheets (CSS)', 'text/css', '.css'), ('ChemDraw eXchange file', 'chemical/x-cdx', '.cdx'), ('Chemical Markup Language', 'chemical/x-cml', '.cml'), ('Chemical Style Markup Language', 'chemical/x-csml', '.csml'), ('CIM Database', 'application/vnd.contact.cmsg', '.cdbcmg'), ('Claymore Data Files', 'application/vnd.claymore', '.cla'), ('Clonk Game', 'application/vnd.clonk.c4group', '.c4g'), ('Close Captioning - Subtitle', 'image/vnd.dvb.subtitle', '.sub'), ('Cloud Data Management Interface (CDMI) - Capability', 'application/cdmi-capability', '.cdmia'), ('Cloud Data Management Interface (CDMI) - Container', 'application/cdmi-container', '.cdmic'), ('Cloud Data Management Interface (CDMI) - Domain', 'application/cdmi-domain', '.cdmid'), ('Cloud Data Management Interface (CDMI) - Object', 'application/cdmi-object', '.cdmio'), ('Cloud Data Management Interface (CDMI) - Queue', 'application/cdmi-queue', '.cdmiq'), ('ClueTrust CartoMobile - Config', 'application/vnd.cluetrust.cartomobile-config', '.c11amc'), ('ClueTrust CartoMobile - Config Package', 'application/vnd.cluetrust.cartomobile-config-pkg', '.c11amz'), ('CMU Image', 'image/x-cmu-raster', '.ras'), ('COLLADA', 'model/vnd.collada+xml', '.dae'), ('Comma-Separated Values', 'text/csv', '.csv'), ('Compact Pro', 'application/mac-compactpro', '.cpt'), ('Compiled Wireless Markup Language (WMLC)', 'application/vnd.wap.wmlc', '.wmlc'), ('Computer Graphics Metafile', 'image/cgm', '.cgm'), ('CoolTalk', 'x-conference/x-cooltalk', '.ice'), ('Corel Metafile Exchange (CMX)', 'image/x-cmx', '.cmx'), ('CorelXARA', 'application/vnd.xara', '.xar'), ('CosmoCaller', 'application/vnd.cosmocaller', '.cmc'), ('CPIO Archive', 'application/x-cpio', '.cpio'), ('CrickSoftware - Clicker', 'application/vnd.crick.clicker', '.clkx'), ('CrickSoftware - Clicker - Keyboard', 'application/vnd.crick.clicker.keyboard', '.clkk'), ('CrickSoftware - Clicker - Palette', 'application/vnd.crick.clicker.palette', '.clkp'), ('CrickSoftware - Clicker - Template', 'application/vnd.crick.clicker.template', '.clkt'), ('CrickSoftware - Clicker - Wordbank', 'application/vnd.crick.clicker.wordbank', '.clkw'), ('Critical Tools - PERT Chart EXPERT', 'application/vnd.criticaltools.wbs+xml', '.wbs'), ('CryptoNote', 'application/vnd.rig.cryptonote', '.cryptonote'),

('Crystallographic Interchange Format', 'chemical/x-cif', '.cif'),  
('CrystalMaker Data Format', 'chemical/x-cmdf', '.cmdf'), ('CU-SeeMe',  
'application/cu-seeme', '.cu'), ('CU-Writer', 'application/prs.cww', '.cww'),  
('Curl - Applet', 'text/vnd.curl', '.curl'), ('Curl - Detached Applet',  
'text/vnd.curl.dcurl', '.dcurl'), ('Curl - Manifest File',  
'text/vnd.curl.mcurl', '.mcurl'), ('Curl - Source Code',  
'text/vnd.curl.scurl', '.scurl'), ('CURL Applet', 'application/vnd.curl.car',  
'.car'), ('CURL Applet', 'application/vnd.curl.pcurl', '.pcurl'),  
('CustomMenu', 'application/vnd.yellowriver-custom-menu', '.cmp'), ('Data  
Structure for the Security Suitability of Cryptographic Algorithms',  
'application/dssc+der', '.dssc'), ('Data Structure for the Security  
Suitability of Cryptographic Algorithms', 'application/dssc+xml', '.xdssc'),  
('Debian Package', 'application/x-debian-package', '.deb'), ('DECE Audio',  
'audio/vnd.dece.audio', '.uva'), ('DECE Graphic', 'image/vnd.dece.graphic',  
'.uvi'), ('DECE High Definition Video', 'video/vnd.dece.hd', '.uvh'), ('DECE  
Mobile Video', 'video/vnd.dece.mobile', '.uvm'), ('DECE MP4',  
'video/vnd.uvu.mp4', '.uvu'), ('DECE PD Video', 'video/vnd.dece.pd',  
'.uvp'), ('DECE SD Video', 'video/vnd.dece.sd', '.uvs'), ('DECE Video',  
'video/vnd.dece.video', '.uvv'), ('Device Independent File Format (DVI)',  
'application/x-dvi', '.dvi'), ('Digital Sismograph Networks - SEED  
Datafiles', 'application/vnd.fdsn.seed', '.seed'), ('Digital Talking Book',  
'application/x-dtbook+xml', '.dtb'), ('Digital Talking Book - Resource File',  
'application/x-dtbresource+xml', '.res'), ('Digital Video Broadcasting',  
'application/vnd.dvb.ait', '.ait'), ('Digital Video Broadcasting',  
'application/vnd.dvb.service', '.svc'), ('Digital Winds Music',  
'audio/vnd.digital-winds', '.eol'), ('DjVu', 'image/vnd.djvu', '.djvu'),  
('Document Type Definition', 'application/xml-dtd', '.dtd'), ('Dolby Meridian  
Lossless Packing', 'application/vnd.dolby.mlp', '.mlp'), ('Doom Video Game',  
'application/x-doom', '.wad'), ('DPGraph', 'application/vnd.dpgraph',  
'.dpg'), ('DRA Audio', 'audio/vnd.dra', '.dra'), ('DreamFactory',  
'application/vnd.dreamfactory', '.dfac'), ('DTS Audio', 'audio/vnd.dts',  
'.dts'), ('DTS High Definition Audio', 'audio/vnd.dts.hd', '.dtshd'), ('DWG  
Drawing', 'image/vnd.dwg', '.dwg'), ('DynaGeo', 'application/vnd.dynageo',  
'.geo'), ('ECMAScript', 'application/ecmascript', '.es'), ('EcoWin Chart',  
'application/vnd.ecowin.chart', '.mag'), ('EDMICS 2000',  
'image/vnd.fujixerox.edmics-mmr', '.mmr'), ('EDMICS 2000',  
'image/vnd.fujixerox.edmics-rlc', '.rlc'), ('Efficient XML Interchange',  
'application/exi', '.exi'), ('EFI Proteus',  
'application/vnd.proteus.magazine', '.mgz'), ('Electronic Publication',  
'application/epub+zip', '.epub'), ('Email Message', 'message/rfc822',  
'.eml'), ('Enliven Viewer', 'application/vnd.enliven', '.nml'), ('Express by  
Infoseek', 'application/vnd.is-xpr', '.xpr'), ('eXtended Image File Format  
(XIFF)', 'image/vnd.xiff', '.xif'), ('Extensible Forms Description Language',  
'application/vnd.xfdl', '.xfdl'), ('Extensible MultiModal Annotation',  
'application/emma+xml', '.emma'), ('EZPix Secure Photo Album',  
'application/vnd.ezpix-album', '.ez2'), ('EZPix Secure Photo Album',

'application/vnd.ezpix-package', '.ez3'), ('FAST Search & Transfer ASA',  
 'image/vnd.fst', '.fst'), ('FAST Search & Transfer ASA', 'video/vnd.fvt',  
 '.fvt'), ('FastBid Sheet', 'image/vnd.fastbidsheet', '.fbs'), ('FCS Express  
 Layout Link', 'application/vnd.denovo.fcselayout-link', '.fe\_launch'),  
 ('Flash Video', 'video/x-f4v', '.f4v'), ('Flash Video', 'video/x-flv',  
 '.flv'), ('FlashPix', 'image/vnd.fpx', '.fpx'), ('FlashPix',  
 'image/vnd.net-fpx', '.npx'), ('FLEXSTOR', 'text/vnd.fmi.flexstor', '.flx'),  
 ('FLI/FLC Animation Format', 'video/x-fli', '.fli'), ('FluxTime Clip',  
 'application/vnd.fluxtime.clip', '.ftc'), ('Forms Data Format',  
 'application/vnd.fdf', '.fdf'), ('Fortran Source File', 'text/x-fortran',  
 '.f'), ('FrameMaker Interchange Format', 'application/vnd.mif', '.mif'),  
 ('FrameMaker Normal Format', 'application/vnd.frameMaker', '.fm'), ('FreeHand  
 MX', 'image/x-freehand', '.fh'), ('Friendly Software Corporation',  
 'application/vnd.fsc.weblaunch', '.fsc'), ('Frogans Player',  
 'application/vnd.frogans.fnc', '.fnc'), ('Frogans Player',  
 'application/vnd.frogans.ltf', '.ltf'), ('Fujitsu - Xerox 2D CAD Data',  
 'application/vnd.fujixerox.ddd', '.ddd'), ('Fujitsu - Xerox DocuWorks',  
 'application/vnd.fujixerox.docuworks', '.xdw'), ('Fujitsu - Xerox DocuWorks  
 Binder', 'application/vnd.fujixerox.docuworks.binder', '.xbd'), ('Fujitsu  
 Oasys', 'application/vnd.fujitsu.oasys', '.oas'), ('Fujitsu Oasys',  
 'application/vnd.fujitsu.oasys2', '.oa2'), ('Fujitsu Oasys',  
 'application/vnd.fujitsu.oasys3', '.oa3'), ('Fujitsu Oasys',  
 'application/vnd.fujitsu.oasysgp', '.fg5'), ('Fujitsu Oasys',  
 'application/vnd.fujitsu.oasysprs', '.bh2'), ('FutureSplash Animator',  
 'application/x-futuresplash', '.spl'), ('FuzzySheet',  
 'application/vnd.fuzzysheet', '.fzs'), ('G3 Fax Image', 'image/g3fax',  
 '.g3'), ('GameMaker ActiveX', 'application/vnd.gmx', '.gmx'), ('Gen-Trix  
 Studio', 'model/vnd.gtw', '.gtw'), ('Genomatix Tuxedo Framework',  
 'application/vnd.genomatix.tuxedo', '.txd'), ('GeoGebra',  
 'application/vnd.geogebra.file', '.ggb'), ('GeoGebra',  
 'application/vnd.geogebra.tool', '.ggt'), ('Geometric Description Language  
 (GDL)', 'model/vnd.gdl', '.gdl'), ('GeoMetry Explorer',  
 'application/vnd.geometry-explorer', '.gex'), ('GEONExT and JSXGraph',  
 'application/vnd.geonext', '.gxt'), ('GeoplanW', 'application/vnd.geoplan',  
 '.g2w'), ('GeospacW', 'application/vnd.geospace', '.g3w'), ('Ghostscript  
 Font', 'application/x-font-ghostscript', '.gsf'), ('Glyph Bitmap Distribution  
 Format', 'application/x-font-bdf', '.bdf'), ('GNU Tar Files',  
 'application/x-gtar', '.gtar'), ('GNU Texinfo Document',  
 'application/x-texinfo', '.texinfo'), ('Gnumeric', 'application/x-gnumeric',  
 '.gnumeric'), ('Google Earth - KML', 'application/vnd.google-earth.kml+xml',  
 '.kml'), ('Google Earth - Zipped KML', 'application/vnd.google-earth.kmz',  
 '.kmz'), ('GrafEq', 'application/vnd.grafeq', '.gqf'), ('Graphics Interchange  
 Format', 'image/gif', '.gif'), ('Graphviz', 'text/vnd.graphviz', '.gv'),  
 ('Groove - Account', 'application/vnd.groove-account', '.gac'), ('Groove -  
 Help', 'application/vnd.groove-help', '.ghf'), ('Groove - Identity Message',  
 'application/vnd.groove-identity-message', '.gim'), ('Groove - Injector',

'application/vnd.groove-injector', '.grv'), ('Groove - Tool Message',  
'application/vnd.groove-tool-message', '.gtm'), ('Groove - Tool Template',  
'application/vnd.groove-tool-template', '.tpl'), ('Groove - Vcard',  
'application/vnd.groove-vcard', '.vcg'), ('H.261', 'video/h261', '.h261'),  
('H.263', 'video/h263', '.h263'), ('H.264', 'video/h264', '.h264'), ('Hewlett  
Packard Instant Delivery', 'application/vnd.hp-hpid', '.hpid'),  
('Hewlett-Packard's WebPrintSmart', 'application/vnd.hp-hps', '.hps'),  
('Hierarchical Data Format', 'application/x-hdf', '.hdf'), ('Hit'n'Mix',  
'audio/vnd.rip', '.rip'), ('Homebanking Computer Interface (HBCI)',  
'application/vnd.hbci', '.hbci'), ('HP Indigo Digital Press - Job Layout  
Language', 'application/vnd.hp-jlyt', '.jlt'), ('HP Printer Command  
Language', 'application/vnd.hp-pcl', '.pcl'), ('HP-GL/2 and HP RTL',  
'application/vnd.hp-hpgl', '.hpgl'), ('HV Script',  
'application/vnd.yamaha.hv-script', '.hvs'), ('HV Voice Dictionary',  
'application/vnd.yamaha.hv-dic', '.hvd'), ('HV Voice Parameter',  
'application/vnd.yamaha.hv-voice', '.hvp'), ('Hydrostatix Master Suite',  
'application/vnd.hydrostatix.sof-data', '.sfd-hdstx'), ('Hyperstudio',  
'application/hyperstudio', '.stk'), ('Hypertext Application Language',  
'application/vnd.hal+xml', '.hal'), ('HyperText Markup Language (HTML)',  
'text/html', '.html'), ('IBM DB2 Rights Manager',  
'application/vnd.ibm.rights-management', '.irm'), ('IBM Electronic Media  
Management System - Secure Container',  
'application/vnd.ibm.secure-container', '.sc'), ('iCalendar',  
'text/calendar', '.ics'), ('ICC profile', 'application/vnd.iccprofile',  
'.icc'), ('Icon Image', 'image/x-icon', '.ico'), ('igLoader',  
'application/vnd.igloader', '.igl'), ('Image Exchange Format', 'image/ief',  
'.ief'), ('ImmerVision PURE Players', 'application/vnd.immervision-ivp',  
'.ivp'), ('ImmerVision PURE Players', 'application/vnd.immervision-ivu',  
'.ivu'), ('IMS Networks', 'application/reginfo+xml', '.rif'), ('In3D - 3DML',  
'text/vnd.in3d.3dml', '.3dml'), ('In3D - 3DML', 'text/vnd.in3d.spot',  
'.spot'), ('Initial Graphics Exchange Specification (IGES)', 'model/iges',  
'.igs'), ('Interactive Geometry Software', 'application/vnd.intergeo',  
'.i2g'), ('Interactive Geometry Software Cinderella',  
'application/vnd.cinderella', '.cdy'), ('Intercon FormNet',  
'application/vnd.intercon.formnet', '.xpw'), ('International Society for  
Advancement of Cytometry', 'application/vnd.isac.fcs', '.fcs'), ('Internet  
Protocol Flow Information Export', 'application/ipfix', '.ipfix'), ('Internet  
Public Key Infrastructure - Certificate', 'application/pkix-cert', '.cer'),  
('Internet Public Key Infrastructure - Certificate Management Protocole',  
'application/pkixcmp', '.pki'), ('Internet Public Key Infrastructure -  
Certificate Revocation Lists', 'application/pkix-crl', '.crl'), ('Internet  
Public Key Infrastructure - Certification Path', 'application/pkix-pkipath',  
'.pkipath'), ('IOCOM Visimeet', 'application/vnd.insors.igm', '.igm'), ('IP  
Unplugged Roaming Client', 'application/vnd.ipunplugged.rcprofile',  
'.rcprofile'), ('iRepository / Lucidoc Editor',  
'application/vnd.irepository.package+xml', '.irp'), ('J2ME App Descriptor',



'text/vnd.sun.j2me.app-descriptor', '.jad'), ('Java Archive',  
 'application/java-archive', '.jar'), ('Java Bytecode File',  
 'application/java-vm', '.class'), ('Java Network Launching Protocol',  
 'application/x-java-jnlp-file', '.jnlp'), ('Java Serialized Object',  
 'application/java-serialized-object', '.ser'), ('Java Source File',  
 'text/x-java-source,java', '.java'), ('JavaScript', 'application/javascript',  
 '.js'), ('JavaScript Object Notation (JSON)', 'application/json', '.json'),  
 ('Joda Archive', 'application/vnd.joost.joda-archive', '.joda'), ('JPEG 2000  
 Compound Image File Format', 'video/jpm', '.jpm'), ('JPEG Image',  
 'image/jpeg', '.jpeg, .jpg'), ('JPGVideo', 'video/jpeg', '.jpgv'),  
 ('Kahootz', 'application/vnd.kahootz', '.ktz'), ('Karaoke on Chipnuts  
 Chipsets', 'application/vnd.chipnuts.karaoke-mmd', '.mmd'), ('KDE KOffice  
 Office Suite - Karbon', 'application/vnd.kde.karbon', '.karbon'), ('KDE  
 KOffice Office Suite - KChart', 'application/vnd.kde.kchart', '.chrt'), ('KDE  
 KOffice Office Suite - Kformula', 'application/vnd.kde.kformula', '.kfo'),  
 ('KDE KOffice Office Suite - Kivio', 'application/vnd.kde.kivio', '.flw'),  
 ('KDE KOffice Office Suite - Kontour', 'application/vnd.kde.kontour',  
 '.kon'), ('KDE KOffice Office Suite - Kpresenter',  
 'application/vnd.kde.kpresenter', '.kpr'), ('KDE KOffice Office Suite -  
 Kspread', 'application/vnd.kde.kspread', '.ksp'), ('KDE KOffice Office Suite  
 - Kword', 'application/vnd.kde.kword', '.kwd'), ('Kenamea App',  
 'application/vnd.kenameaapp', '.htke'), ('Kidspiration',  
 'application/vnd.kidspiration', '.kia'), ('Kinar Applications',  
 'application/vnd.kinar', '.kne'), ('Kodak Storyshare',  
 'application/vnd.kodak-descriptor', '.sse'), ('Laser App Enterprise',  
 'application/vnd.las.las+xml', '.lasxml'), ('LaTeX', 'application/x-latex',  
 '.latex'), ('Life Balance - Desktop Edition',  
 'application/vnd.llamagraphics.life-balance.desktop', '.lbd'), ('Life Balance  
 - Exchange Format',  
 'application/vnd.llamagraphics.life-balance.exchange+xml', '.lbe'),  
 ('Lightspeed Audio Lab', 'application/vnd.jam', '.jam'), ('Lotus 1-2-3',  
 'application/vnd.lotus-1-2-3', '.123'), ('Lotus Approach',  
 'application/vnd.lotus-approach', '.apr'), ('Lotus Freelance',  
 'application/vnd.lotus-freelance', '.pre'), ('Lotus Notes',  
 'application/vnd.lotus-notes', '.nsf'), ('Lotus Organizer',  
 'application/vnd.lotus-organizer', '.org'), ('Lotus Screencam',  
 'application/vnd.lotus-screencam', '.scm'), ('Lotus Wordpro',  
 'application/vnd.lotus-wordpro', '.lwp'), ('Lucent Voice',  
 'audio/vnd.lucent.voice', '.lvp'), ('M3U (Multimedia Playlist)',  
 'audio/x-mpegurl', '.m3u'), ('M4v', 'video/x-m4v', '.m4v'), ('Macintosh  
 BinHex 4.0', 'application/mac-binhex40', '.hqx'), ('MacPorts Port System',  
 'application/vnd.macports.portpkg', '.portpkg'), ('MapGuide DBXML',  
 'application/vnd.osgeo.mapguide.package', '.mgp'), ('MARC Formats',  
 'application/marc', '.mrc'), ('MARC21 XML Schema', 'application/marcxml+xml',  
 '.mrcx'), ('Material Exchange Format', 'application/mxf', '.mxf'),  
 ('Mathematica Notebook Player', 'application/vnd.wolfram.player', '.nbp'),

('Mathematica Notebooks', 'application/mathematica', '.ma'), ('Mathematical Markup Language', 'application/mathml+xml', '.mathml'), ('Mbox database files', 'application/mbox', '.mbox'), ('MedCalc', 'application/vnd.medcalcddata', '.mc1'), ('Media Server Control Markup Language', 'application/mediaservercontrol+xml', '.mscml'), ('MediaRemote', 'application/vnd.mediastation.cdkey', '.cdkey'), ('Medical Waveform Encoding Format', 'application/vnd.mfer', '.mwf'), ('Melody Format for Mobile Platform', 'application/vnd.mfmp', '.mfmp'), ('Mesh Data Type', 'model/mesh', '.msh'), ('Metadata Authority Description Schema', 'application/mads+xml', '.mads'), ('Metadata Encoding and Transmission Standard', 'application/mets+xml', '.mets'), ('Metadata Object Description Schema', 'application/mods+xml', '.mods'), ('Metalink', 'application/metalink4+xml', '.meta4'), ('Microsoft PowerPoint - Macro-Enabled Template File', 'application/vnd.ms-powerpoint.template.macroenabled.12', '.potm'), ('Microsoft Word - Macro-Enabled Document', 'application/vnd.ms-word.document.macroenabled.12', '.docm'), ('Microsoft Word - Macro-Enabled Template', 'application/vnd.ms-word.template.macroenabled.12', '.dotm'), ('Micro CADAM Helix D&D', 'application/vnd.mcd', '.mcd'), ('Micrografx', 'application/vnd.micrografx.flo', '.flo'), ('Micrografx iGrafX Professional', 'application/vnd.micrografx.igx', '.igx'), ('MICROSEC e-Szign', 'application/vnd.eszigno3+xml', '.es3'), ('Microsoft Access', 'application/x-msaccess', '.mdb'), ('Microsoft Advanced Systems Format (ASF)', 'video/x-ms-asf', '.asf'), ('Microsoft Application', 'application/x-msdownload', '.exe'), ('Microsoft Artgalry', 'application/vnd.ms-artgalry', '.cil'), ('Microsoft Cabinet File', 'application/vnd.ms-cab-compressed', '.cab'), ('Microsoft Class Server', 'application/vnd.ms-ims', '.ims'), ('Microsoft ClickOnce', 'application/x-ms-application', '.application'), ('Microsoft Clipboard Clip', 'application/x-msclip', '.clp'), ('Microsoft Document Imaging Format', 'image/vnd.ms-modi', '.mdi'), ('Microsoft Embedded OpenType', 'application/vnd.ms-fontobject', '.eot'), ('Microsoft Excel', 'application/vnd.ms-excel', '.xls'), ('Microsoft Excel - Add-In File', 'application/vnd.ms-excel.addin.macroenabled.12', '.xlam'), ('Microsoft Excel - Binary Workbook', 'application/vnd.ms-excel.sheet.binary.macroenabled.12', '.xlsb'), ('Microsoft Excel - Macro-Enabled Template File', 'application/vnd.ms-excel.template.macroenabled.12', '.xltm'), ('Microsoft Excel - Macro-Enabled Workbook', 'application/vnd.ms-excel.sheet.macroenabled.12', '.xlsm'), ('Microsoft Html Help File', 'application/vnd.ms-htmlhelp', '.chm'), ('Microsoft Information Card', 'application/x-mscardfile', '.crd'), ('Microsoft Learning Resource Module', 'application/vnd.ms-lrm', '.lrm'), ('Microsoft MediaView', 'application/x-msmediaview', '.mvb'), ('Microsoft Money', 'application/x-msmoney', '.mny'), ('Microsoft Office - OOXML - Presentation', 'application/vnd.openxmlformats-officedocument.presentationml.presentation', '.pptx'), ('Microsoft Office - OOXML - Presentation (Slide)',

'application/vnd.openxmlformats-officedocument.presentationml.slide',  
'.sldx'), ('Microsoft Office - OOXML - Presentation (Slideshow)',  
'application/vnd.openxmlformats-officedocument.presentationml.slideshow',  
'.ppsx'), ('Microsoft Office - OOXML - Presentation Template',  
'application/vnd.openxmlformats-officedocument.presentationml.template',  
'.potx'), ('Microsoft Office - OOXML - Spreadsheet',  
'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',  
'.xlsx'), ('Microsoft Office - OOXML - Spreadsheet Teplate',  
'application/vnd.openxmlformats-officedocument.spreadsheetml.template',  
'.xltx'), ('Microsoft Office - OOXML - Word Document',  
'application/vnd.openxmlformats-officedocument.wordprocessingml.document',  
'.docx'), ('Microsoft Office - OOXML - Word Document Template',  
'application/vnd.openxmlformats-officedocument.wordprocessingml.template',  
'.dotx'), ('Microsoft Office Binder', 'application/x-msbinder', '.obd'),  
('Microsoft Office System Release Theme', 'application/vnd.ms-officetheme',  
'.thmx'), ('Microsoft OneNote', 'application/onenote', '.onetoc'),  
('Microsoft PlayReady Ecosystem', 'audio/vnd.ms-playready.media.pya',  
'.pya'), ('Microsoft PlayReady Ecosystem Video',  
'video/vnd.ms-playready.media.pyv', '.pyv'), ('Microsoft PowerPoint',  
'application/vnd.ms-powerpoint', '.ppt'), ('Microsoft PowerPoint - Add-in  
file', 'application/vnd.ms-powerpoint.addin.macroenabled.12', '.ppam'),  
('Microsoft PowerPoint - Macro-Enabled Open XML Slide',  
'application/vnd.ms-powerpoint.slide.macroenabled.12', '.sldm'), ('Microsoft  
PowerPoint - Macro-Enabled Presentation File',  
'application/vnd.ms-powerpoint.presentation.macroenabled.12', '.pptm'),  
('Microsoft PowerPoint - Macro-Enabled Slide Show File',  
'application/vnd.ms-powerpoint.slideshow.macroenabled.12', '.ppsm'),  
('Microsoft Project', 'application/vnd.ms-project', '.mpp'), ('Microsoft  
Publisher', 'application/x-mspublisher', '.pub'), ('Microsoft Schedule+',  
'application/x-msschedule', '.scd'), ('Microsoft Silverlight',  
'application/x-silverlight-app', '.xap'), ('Microsoft Trust UI Provider -  
Certificate Trust Link', 'application/vnd.ms-pki.stl', '.stl'), ('Microsoft  
Trust UI Provider - Security Catalog', 'application/vnd.ms-pki.seccat',  
'.cat'), ('Microsoft Visio', 'application/vnd.visio', '.vsd'), ('Microsoft  
Windows Media', 'video/x-ms-wm', '.wm'), ('Microsoft Windows Media Audio',  
'audio/x-ms-wma', '.wma'), ('Microsoft Windows Media Audio Redirector',  
'audio/x-ms-wax', '.wax'), ('Microsoft Windows Media Audio/Video Playlist',  
'video/x-ms-wmx', '.wmx'), ('Microsoft Windows Media Player Download  
Package', 'application/x-ms-wmd', '.wmd'), ('Microsoft Windows Media Player  
Playlist', 'application/vnd.ms-wpl', '.wpl'), ('Microsoft Windows Media  
Player Skin Package', 'application/x-ms-wmz', '.wmz'), ('Microsoft Windows  
Media Video', 'video/x-ms-wmv', '.wmv'), ('Microsoft Windows Media Video  
Playlist', 'video/x-ms-wvx', '.wvx'), ('Microsoft Windows Metafile',  
'application/x-msmetafile', '.wmf'), ('Microsoft Windows Terminal Services',  
'application/x-msterminal', '.trm'), ('Microsoft Word', 'application/msword',  
'.doc'), ('Microsoft Wordpad', 'application/x-mswrite', '.wri'), ('Microsoft

Works', 'application/vnd.ms-works', '.wps'), ('Microsoft XAML Browser Application', 'application/x-ms-xbap', '.xbap'), ('Microsoft XML Paper Specification', 'application/vnd.ms-xpsdocument', '.xps'), ('MIDI - Musical Instrument Digital Interface', 'audio/midi', '.mid'), ('MiniPay', 'application/vnd.ibm.minipay', '.mpy'), ('MO:DCA-P', 'application/vnd.ibm.modcap', '.afp'), ('Mobile Information Device Profile', 'application/vnd.jcp.javame.midlet-rms', '.rms'), ('MobileTV', 'application/vnd.tmobile-livetv', '.tmo'), ('Mobipocket', 'application/x-mobipocket-ebook', '.prc'), ('Mobius Management Systems - Basket file', 'application/vnd.mobius.mbk', '.mbk'), ('Mobius Management Systems - Distribution Database', 'application/vnd.mobius.dis', '.dis'), ('Mobius Management Systems - Policy Definition Language File', 'application/vnd.mobius.plc', '.plc'), ('Mobius Management Systems - Query File', 'application/vnd.mobius.mqy', '.mqy'), ('Mobius Management Systems - Script Language', 'application/vnd.mobius.msl', '.msl'), ('Mobius Management Systems - Topic Index File', 'application/vnd.mobius.txf', '.txf'), ('Mobius Management Systems - UniversalArchive', 'application/vnd.mobius.daf', '.daf'), ('mod\_fly / fly.cgi', 'text/vnd.fly', '.fly'), ('Mophun Certificate', 'application/vnd.mophun.certificate', '.mpc'), ('Mophun VM', 'application/vnd.mophun.application', '.mpn'), ('Motion JPEG 2000', 'video/mj2', '.mj2'), ('MPEG Audio', 'audio/mpeg', '.mpga'), ('MPEG Url', 'video/vnd.mpegurl', '.mxu'), ('MPEG Video', 'video/mpeg', '.mpeg'), ('MPEG-21', 'application/mp21', '.m21'), ('MPEG-4 Audio', 'audio/mp4', '.mp4a'), ('MPEG-4 Video', 'video/mp4', '.mp4'), ('MPEG4', 'application/mp4', '.mp4'), ('Multimedia Playlist Unicode', 'application/vnd.apple.mpegurl', '.m3u8'), ('Musical Score Interpreted Code Invented for the ASCII designation of Notation', 'application/vnd.musician', '.mus'), ('Muvee Automatic Video Editing', 'application/vnd.muvee.style', '.msty'), ('MXML', 'application/xv+xml', '.mxml'), ('N-Gage Game Data', 'application/vnd.nokia.n-gage.data', '.ngdat'), ('N-Gage Game Installer', 'application/vnd.nokia.n-gage.symbian.install', '.n-gage'), ('Navigation Control file for XML (for ePub)', 'application/x-dtbncx+xml', '.ncx'), ('Network Common Data Form (NetCDF)', 'application/x-netcdf', '.nc'), ('neuroLanguage', 'application/vnd.neurolanguage.nlu', '.nlu'), ('New Moon Liftoff/DNA', 'application/vnd.dna', '.dna'), ('NobleNet Directory', 'application/vnd.noblenet-directory', '.nnd'), ('NobleNet Sealer', 'application/vnd.noblenet-sealer', '.nns'), ('NobleNet Web', 'application/vnd.noblenet-web', '.nnw'), ('Nokia Radio Application - Preset', 'application/vnd.nokia.radio-preset', '.rpst'), ('Nokia Radio Application - Preset', 'application/vnd.nokia.radio-presets', '.rpss'), ('Notation3', 'text/n3', '.n3'), ('Novadigm's RADIA and EDM products', 'application/vnd.novadigm.edm', '.edm'), ('Novadigm's RADIA and EDM products', 'application/vnd.novadigm.edx', '.edx'), ('Novadigm's RADIA and EDM products', 'application/vnd.novadigm.ext', '.ext'), ('NpGraphIt', 'application/vnd.flographit', '.gph'), ('Nuera ECELP 4800', 'audio/vnd.nuera.ecelp4800', '.ecelp4800'), ('Nuera ECELP 7470',

'audio/vnd.nuera.ecelp7470', '.ecelp7470'), ('Nuera ECELP 9600',  
 'audio/vnd.nuera.ecelp9600', '.ecelp9600'), ('Office Document Architecture',  
 'application/oda', '.oda'), ('Ogg', 'application/ogg', '.ogx'), ('Ogg Audio',  
 'audio/ogg', '.oga'), ('Ogg Video', 'video/ogg', '.ogv'), ('OMA Download  
 Agents', 'application/vnd.oma.dd2+xml', '.dd2'), ('Open Document Text Web',  
 'application/vnd.oasis.opendocument.text-web', '.oth'), ('Open eBook  
 Publication Structure', 'application/oebps-package+xml', '.opf'), ('Open  
 Financial Exchange', 'application/vnd.intu.qbo', '.qbo'), ('Open Office  
 Extension', 'application/vnd.openofficeorg.extension', '.oxt'), ('Open Score  
 Format', 'application/vnd.yamaha.openscoreformat', '.osf'), ('Open Web Media  
 Project - Audio', 'audio/webm', '.weba'), ('Open Web Media Project - Video',  
 'video/webm', '.webm'), ('OpenDocument Chart',  
 'application/vnd.oasis.opendocument.chart', '.odc'), ('OpenDocument Chart  
 Template', 'application/vnd.oasis.opendocument.chart-template', '.otc'),  
 ('OpenDocument Database', 'application/vnd.oasis.opendocument.database',  
 '.odb'), ('OpenDocument Formula',  
 'application/vnd.oasis.opendocument.formula', '.odf'), ('OpenDocument Formula  
 Template', 'application/vnd.oasis.opendocument.formula-template', '.odft'),  
 ('OpenDocument Graphics', 'application/vnd.oasis.opendocument.graphics',  
 '.odg'), ('OpenDocument Graphics Template',  
 'application/vnd.oasis.opendocument.graphics-template', '.otg'),  
 ('OpenDocument Image', 'application/vnd.oasis.opendocument.image', '.odi'),  
 ('OpenDocument Image Template',  
 'application/vnd.oasis.opendocument.image-template', '.oti'), ('OpenDocument  
 Presentation', 'application/vnd.oasis.opendocument.presentation', '.odp'),  
 ('OpenDocument Presentation Template',  
 'application/vnd.oasis.opendocument.presentation-template', '.otp'),  
 ('OpenDocument Spreadsheet',  
 'application/vnd.oasis.opendocument.spreadsheet', '.ods'), ('OpenDocument  
 Spreadsheet Template',  
 'application/vnd.oasis.opendocument.spreadsheet-template', '.ots'),  
 ('OpenDocument Text', 'application/vnd.oasis.opendocument.text', '.odt'),  
 ('OpenDocument Text Master',  
 'application/vnd.oasis.opendocument.text-master', '.odm'), ('OpenDocument  
 Text Template', 'application/vnd.oasis.opendocument.text-template', '.ott'),  
 ('OpenGL Textures (KTX)', 'image/ktx', '.ktx'), ('OpenOffice - Calc  
 (Spreadsheet)', 'application/vnd.sun.xml.calc', '.sxc'), ('OpenOffice - Calc  
 Template (Spreadsheet)', 'application/vnd.sun.xml.calc.template', '.stc'),  
 ('OpenOffice - Draw (Graphics)', 'application/vnd.sun.xml.draw', '.sxd'),  
 ('OpenOffice - Draw Template (Graphics)',  
 'application/vnd.sun.xml.draw.template', '.std'), ('OpenOffice - Impress  
 (Presentation)', 'application/vnd.sun.xml.impress', '.sxi'), ('OpenOffice -  
 Impress Template (Presentation)', 'application/vnd.sun.xml.impress.template',  
 '.sti'), ('OpenOffice - Math (Formula)', 'application/vnd.sun.xml.math',  
 '.sxm'), ('OpenOffice - Writer (Text - HTML)',  
 'application/vnd.sun.xml.writer', '.sxw'), ('OpenOffice - Writer (Text -

HTML)', 'application/vnd.sun.xml.writer.global', '.sxc'), ('OpenOffice - Writer Template (Text - HTML)', 'application/vnd.sun.xml.writer.template', '.stw'), ('OpenType Font File', 'application/x-font-otf', '.otf'), ('OSFPVG', 'application/vnd.yamaha.openscoreformat.osfpvg+xml', '.osfpvg'), ('OSGi Deployment Package', 'application/vnd.osgi.dp', '.dp'), ('PalmOS Data', 'application/vnd.palm', '.pdb'), ('Pascal Source File', 'text/x-pascal', '.p'), ('PawaaFILE', 'application/vnd.pawaafile', '.paw'), ('PCL 6 Enhanced (Formerly PCL XL)', 'application/vnd.hp-pclxl', '.pclxl'), ('Pcset eFIF File', 'application/vnd.picset', '.efif'), ('PCX Image', 'image/x-pcx', '.pcx'), ('Photoshop Document', 'image/vnd.adobe.photoshop', '.psd'), ('PICSRules', 'application/pics-rules', '.prf'), ('PICT Image', 'image/x-pict', '.pic'), ('pIRCh', 'application/x-chat', '.chat'), ('PKCS #10 - Certification Request Standard', 'application/pkcs10', '.p10'), ('PKCS #12 - Personal Information Exchange Syntax Standard', 'application/x-pkcs12', '.p12'), ('PKCS #7 - Cryptographic Message Syntax Standard', 'application/pkcs7-mime', '.p7m'), ('PKCS #7 - Cryptographic Message Syntax Standard', 'application/pkcs7-signature', '.p7s'), ('PKCS #7 - Cryptographic Message Syntax Standard (Certificate Request Response)', 'application/x-pkcs7-certreqresp', '.p7r'), ('PKCS #7 - Cryptographic Message Syntax Standard (Certificates)', 'application/x-pkcs7-certificates', '.p7b'), ('PKCS #8 - Private-Key Information Syntax Standard', 'application/pkcs8', '.p8'), ('PocketLearn Viewers', 'application/vnd.pocketlearn', '.plf'), ('Portable Anymap Image', 'image/x-portable-anymap', '.pnm'), ('Portable Bitmap Format', 'image/x-portable-bitmap', '.pbm'), ('Portable Compiled Format', 'application/x-font-pcf', '.pcf'), ('Portable Font Resource', 'application/font-tdpfr', '.pfr'), ('Portable Game Notation (Chess Games)', 'application/x-chess-pgn', '.pgn'), ('Portable Graymap Format', 'image/x-portable-graymap', '.pgm'), ('Portable Network Graphics (PNG)', 'image/png', '.png'), ('Portable Pixmap Format', 'image/x-portable-pixmap', '.ppm'), ('Portable Symmetric Key Container', 'application/pskc+xml', '.pskcxml'), ('PosML', 'application/vnd.ctc-posml', '.pml'), ('PostScript', 'application/postscript', '.ai'), ('PostScript Fonts', 'application/x-font-type1', '.pfa'), ('PowerBuilder', 'application/vnd.powerbuilder6', '.pbd'), ('Pretty Good Privacy', 'application/pgp-encrypted', ''), ('Pretty Good Privacy - Signature', 'application/pgp-signature', '.pgp'), ('Preview Systems ZipLock/VBox', 'application/vnd.previewsystems.box', '.box'), ('Princeton Video Image', 'application/vnd.pvi.ptid1', '.ptid'), ('Pronunciation Lexicon Specification', 'application/pls+xml', '.pls'), ('Proprietary P&G Standard Reporting System', 'application/vnd.pg.format', '.str'), ('Proprietary P&G Standard Reporting System', 'application/vnd.pg.osasli', '.ei6'), ('PRS Lines Tag', 'text/prs.lines.tag', '.dsc'), ('PSF Fonts', 'application/x-font-linux-psf', '.psf'), ('PubliShare Objects', 'application/vnd.publishare-delta-tree', '.qps'), ('Qualcomm's Plaza Mobile Internet', 'application/vnd.pmi.widget', '.wg'), ('QuarkXpress', 'application/vnd.quark.quarkxpress', '.qxd'), ('QUASS Stream Player',

'application/vnd.epson.esf', '.esf'), ('QUASS Stream Player',  
'application/vnd.epson.msf', '.msf'), ('QUASS Stream Player',  
'application/vnd.epson.ssf', '.ssf'), ('QuickAnime Player',  
'application/vnd.epson.quickanime', '.qam'), ('Quicken',  
'application/vnd.intu.qfx', '.qfx'), ('Quicktime Video', 'video/quicktime',  
'.qt'), ('RAR Archive', 'application/x-rar-compressed', '.rar'), ('Real Audio  
Sound', 'audio/x-pn-realaudio', '.ram'), ('Real Audio Sound',  
'audio/x-pn-realaudio-plugin', '.rmp'), ('Really Simple Discovery',  
'application/rsd+xml', '.rsd'), ('RealMedia', 'application/vnd.rn-realmedia',  
'.rm'), ('RealVNC', 'application/vnd.realvnc.bed', '.bed'), ('Recordare  
Applications', 'application/vnd.recordare.musicxml', '.mxl'), ('Recordare  
Applications', 'application/vnd.recordare.musicxml+xml', '.musicxml'),  
('Relax NG Compact Syntax', 'application/relax-ng-compact-syntax', '.rnc'),  
('RemoteDocs R-Viewer', 'application/vnd.data-vision.rdz', '.rdz'),  
('Resource Description Framework', 'application/rdf+xml', '.rdf'),  
('RetroPlatform Player', 'application/vnd.cloanto.rp9', '.rp9'), ('RhymBox',  
'application/vnd.jisp', '.jisp'), ('Rich Text Format', 'application/rtf',  
'.rtf'), ('Rich Text Format (RTF)', 'text/richtext', '.rtx'), ('ROUTE 66  
Location Based Services', 'application/vnd.route66.link66+xml', '.link66'),  
('RSS - Really Simple Syndication', 'application/rss+xml', '.rss, .xml'), ('S  
Hexdump Format', 'application/shf+xml', '.shf'), ('SailingTracker',  
'application/vnd.sailingtracker.track', '.st'), ('Scalable Vector Graphics  
(SVG)', 'image/svg+xml', '.svg'), ('ScheduleUs',  
'application/vnd.sus-calendar', '.sus'), ('Search/Retrieve via URL Response  
Format', 'application/sru+xml', '.sru'), ('Secure Electronic Transaction -  
Payment', 'application/set-payment-initiation', '.setpay'), ('Secure  
Electronic Transaction - Registration',  
'application/set-registration-initiation', '.setreg'), ('Secured eMail',  
'application/vnd.sema', '.sema'), ('Secured eMail', 'application/vnd.semd',  
'.semd'), ('Secured eMail', 'application/vnd.semif', '.semif'), ('SeeMail',  
'application/vnd.seemail', '.see'), ('Server Normal Format',  
'application/x-font-snf', '.snf'), ('Server-Based Certificate Validation  
Protocol - Validation Policies - Request', 'application/scvp-vp-request',  
'.spq'), ('Server-Based Certificate Validation Protocol - Validation Policies  
- Response', 'application/scvp-vp-response', '.spp'), ('Server-Based  
Certificate Validation Protocol - Validation Request',  
'application/scvp-cv-request', '.scq'), ('Server-Based Certificate Validation  
Protocol - Validation Response', 'application/scvp-cv-response', '.scs'),  
('Session Description Protocol', 'application/sdp', '.sdp'), ('Setext',  
'text/x-setext', '.etx'), ('SGI Movie', 'video/x-sgi-movie', '.movie'),  
('Shana Informed Filler', 'application/vnd.shana.informed.formdata', '.ifm'),  
('Shana Informed Filler', 'application/vnd.shana.informed.formtemplate',  
'.itp'), ('Shana Informed Filler',  
'application/vnd.shana.informed.interchange', '.iif'), ('Shana Informed  
Filler', 'application/vnd.shana.informed.package', '.ipk'), ('Sharing  
Transaction Fraud Data', 'application/thraud+xml', '.tfi'), ('Shell Archive',

'application/x-shar', '.shar'), ('Silicon Graphics RGB Bitmap',  
'image/x-rgb', '.rgb'), ('SimpleAnimeLite Player',  
'application/vnd.epson.salt', '.slt'), ('Simply Accounting',  
'application/vnd.accpac.simply.aso', '.aso'), ('Simply Accounting - Data  
Import', 'application/vnd.accpac.simply.imp', '.imp'), ('SimTech MindMapper',  
'application/vnd.simtech-mindmapper', '.twd'), ('Sixth Floor Media -  
CommonSpace', 'application/vnd.commonspace', '.csp'), ('SMAF Audio',  
'application/vnd.yamaha.smaf-audio', '.saf'), ('SMAF File',  
'application/vnd.smaf', '.mmf'), ('SMAF Phrase',  
'application/vnd.yamaha.smaf-phrase', '.spf'), ('SMART Technologies Apps',  
'application/vnd.smart.teacher', '.teacher'), ('SourceView Document',  
'application/vnd.svd', '.svd'), ('SPARQL - Query',  
'application/sparql-query', '.rq'), ('SPARQL - Results',  
'application/sparql-results+xml', '.srx'), ('Speech Recognition Grammar  
Specification', 'application/srgs', '.gram'), ('Speech Recognition Grammar  
Specification - XML', 'application/srgs+xml', '.grxml'), ('Speech Synthesis  
Markup Language', 'application/ssml+xml', '.ssml'), ('SSEYO Koan Play File',  
'application/vnd.koan', '.skp'), ('Standard Generalized Markup Language  
(SGML)', 'text/sgml', '.sgml'), ('StarOffice - Calc',  
'application/vnd.stardivision.calc', '.sdc'), ('StarOffice - Draw',  
'application/vnd.stardivision.draw', '.sda'), ('StarOffice - Impress',  
'application/vnd.stardivision.impress', '.sdd'), ('StarOffice - Math',  
'application/vnd.stardivision.math', '.smf'), ('StarOffice - Writer',  
'application/vnd.stardivision.writer', '.sdw'), ('StarOffice - Writer  
(Global)', 'application/vnd.stardivision.writer-global', '.sgl'),  
('StepMania', 'application/vnd.stepmania.stepchart', '.sm'), ('Stuffit  
Archive', 'application/x-stuffit', '.sit'), ('Stuffit Archive',  
'application/x-stuffitx', '.sitx'), ('SudokuMagic',  
'application/vnd.solent.sdkm+xml', '.sdkm'), ('Sugar Linux Application  
Bundle', 'application/vnd.olpc-sugar', '.xo'), ('Sun Audio - Au file format',  
'audio/basic', '.au'), ('SundaHus WQ', 'application/vnd.wqd', '.wqd'),  
('Symbian Install Package', 'application/vnd.symbian.install', '.sis'),  
('Synchronized Multimedia Integration Language', 'application/smil+xml',  
'.smi'), ('SyncML', 'application/vnd.syncml+xml', '.xsm'), ('SyncML - Device  
Management', 'application/vnd.syncml.dm+wbxml', '.bdm'), ('SyncML - Device  
Management', 'application/vnd.syncml.dm+xml', '.xdm'), ('System V Release 4  
CPIO Archive', 'application/x-sv4cpio', '.sv4cpio'), ('System V Release 4  
CPIO Checksum Data', 'application/x-sv4crc', '.sv4crc'), ('Systems Biology  
Markup Language', 'application/sbml+xml', '.sbml'), ('Tab Separated Values',  
'text/tab-separated-values', '.tsv'), ('Tagged Image File Format',  
'image/tiff', '.tiff'), ('Tao Intent',  
'application/vnd.tao.intent-module-archive', '.tao'), ('Tar File (Tape  
Archive)', 'application/x-tar', '.tar'), ('Tcl Script', 'application/x-tcl',  
'.tcl'), ('TeX', 'application/x-tex', '.tex'), ('TeX Font Metric',  
'application/x-tex-tfm', '.tfm'), ('Text Encoding and Interchange',  
'application/tei+xml', '.tei'), ('Text File', 'text/plain', '.txt'), ('TIBCO



Spotfire', 'application/vnd.spotfire.dxp', '.dxp'), ('TIBCO Spotfire',  
 'application/vnd.spotfire.sfs', '.sfs'), ('Time Stamped Data Envelope',  
 'application/timestamped-data', '.tsd'), ('TRI Systems Config',  
 'application/vnd.trid.tpt', '.tpt'), ('Triscape Map Explorer',  
 'application/vnd.triscape.mxs', '.mxs'), ('troff', 'text/troff', '.t'),  
 ('True BASIC', 'application/vnd.trueapp', '.tra'), ('TrueType Font',  
 'application/x-font-ttf', '.ttf'), ('Turtle (Terse RDF Triple Language)',  
 'text/turtle', '.ttl'), ('UMAJIN', 'application/vnd.umajin', '.umj'),  
 ('Unique Object Markup Language', 'application/vnd.uoml+xml', '.uoml'),  
 ('Unity 3d', 'application/vnd.unity', '.unityweb'), ('Universal Forms  
 Description Language', 'application/vnd.ufdl', '.ufd'), ('URI Resolution  
 Services', 'text/uri-list', '.uri'), ('User Interface Quartz - Theme  
 (Symbian)', 'application/vnd.uiq.theme', '.utz'), ('Ustar (Uniform Standard  
 Tape Archive)', 'application/x-ustar', '.ustar'), ('UUEncode',  
 'text/x-uencode', '.uu'), ('vCalendar', 'text/x-vcalendar', '.vcs'),  
 ('vCard', 'text/x-vcard', '.vcf'), ('Video CD', 'application/x-cdlink',  
 '.vcd'), ('Viewport+', 'application/vnd.vsf', '.vsf'), ('Virtual Reality  
 Modeling Language', 'model/vrml', '.wrl'), ('VirtualCatalog',  
 'application/vnd.vcx', '.vcx'), ('Virtue MTS', 'model/vnd.mts', '.mts'),  
 ('Virtue VTU', 'model/vnd.vtu', '.vtu'), ('Visionary',  
 'application/vnd.visionary', '.vis'), ('Vivo', 'video/vnd.vivo', '.viv'),  
 ('Voice Browser Call Control', 'application/ccxml+xml', '.ccxml'),  
 ('VoiceXML', 'application/voicexml+xml', '.vxml'), ('WAIS Source',  
 'application/x-wais-source', '.src'), ('WAP Binary XML (WBXML)',  
 'application/vnd.wap.wbxml', '.wbxml'), ('WAP Bitamp (WBMP)',  
 'image/vnd.wap.wbmp', '.wbmp'), ('Waveform Audio File Format (WAV)',  
 'audio/x-wav', '.wav'), ('Web Distributed Authoring and Versioning',  
 'application/davmount+xml', '.davmount'), ('Web Open Font Format',  
 'application/x-font-woff', '.woff'), ('Web Services Policy',  
 'application/wspolicy+xml', '.wspolicy'), ('WebP Image', 'image/webp',  
 '.webp'), ('WebTurbo', 'application/vnd.webturbo', '.wtb'), ('Widget  
 Packaging and XML Configuration', 'application/widget', '.wgt'), ('WinHelp',  
 'application/winhelp', '.hlp'), ('Wireless Markup Language (WML)',  
 'text/vnd.wap.wml', '.wml'), ('Wireless Markup Language Script (WMLScript)',  
 'text/vnd.wap.wmlscript', '.wmls'), ('WMLScript',  
 'application/vnd.wap.wmlscriptc', '.wmlsc'), ('Wordperfect',  
 'application/vnd.wordperfect', '.wpd'), ('Worldtalk',  
 'application/vnd.wt.stf', '.stf'), ('WSDL - Web Services Description  
 Language', 'application/wsdl+xml', '.wsdl'), ('X BitMap', 'image/x-xbitmap',  
 '.xbm'), ('X PixMap', 'image/x-xpixmap', '.xpm'), ('X Window Dump',  
 'image/x-xwindowdump', '.xwd'), ('X.509 Certificate',  
 'application/x-x509-ca-cert', '.der'), ('Xfig', 'application/x-xfig',  
 '.fig'), ('XHTML - The Extensible HyperText Markup Language',  
 'application/xhtml+xml', '.xhtml'), ('XML - Extensible Markup Language',  
 'application/xml', '.xml'), ('XML Configuration Access Protocol - XCAP Diff',  
 'application/xcap-diff+xml', '.xdf'), ('XML Encryption Syntax and

```

Processing', 'application/xenc+xml', '.xenc'), ('XML Patch Framework',
'application/patch-ops-error+xml', '.xer'), ('XML Resource Lists',
'application/resource-lists+xml', '.rl'), ('XML Resource Lists',
'application/rls-services+xml', '.rs'), ('XML Resource Lists Diff',
'application/resource-lists-diff+xml', '.rld'), ('XML Transformations',
'application/xslt+xml', '.xslt'), ('XML-Binary Optimized Packaging',
'application/xop+xml', '.xop'), ('XPIInstall - Mozilla',
'application/x-xpinstall', '.xpi'), ('XSPF - XML Shareable Playlist Format',
'application/xspf+xml', '.xspf'), ('XUL - XML User Interface Language',
'application/vnd.mozilla.xul+xml', '.xul'), ('XYZ File Format',
'chemical/x-xyz', '.xyz'), ('YANG Data Modeling Language',
'application/yang', '.yang'), ('YIN (YANG - XML)', 'application/yin+xml',
'.yin'), ('Z.U.L. Geometry', 'application/vnd.zul', '.zir'), ('Zip Archive',
'application/zip', '.zip'), ('ZVUE Media Manager',
'application/vnd.handheld-entertainment+xml', '.zmm'), ('Zzazz Deck',
'application/vnd.zzazz.deck+xml', '.zaz') );

```

## 6.3 Variables

### DefInt

---

Declaration DefInt: Integer = 0;

### DefInt64

---

Declaration DefInt64: Int64 = 0;

### DefBool

---

Declaration DefBool: Boolean = False;

### DefTime

---

Declaration DefTime: TTime = NullTime;

### DefDate

---

Declaration DefDate: TDate = NullDate;

### DefDateTime

---

Declaration DefDateTime: TDateTime = NullDateTime;

### DefFloat

---

Declaration DefFloat: Double = 0;

**BROOK\_HTTP\_DAEMON\_DESCRIPTION** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_DAEMON\_DESCRIPTION: string = 'Brook HTTP daemon.';

**BROOK\_HTTP\_DAEMON\_DISPLAYNAME** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_DAEMON\_DISPLAYNAME: string = 'Brook daemon';

**BROOK\_HTTP\_DAEMON\_NAME** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_DAEMON\_NAME: string = 'BrookHttpDaemon';

**BROOK\_HTTP\_DAEMON\_CLASSNAME** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_DAEMON\_CLASSNAME: string = 'TBrookHttpDaemon';

# Chapter 7

## Unit BrookException

### 7.1 Description

Brook framework, Exceptions Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 7.2 Uses

- BrookConsts(6)
- SysUtils

### 7.3 Overview

EBrook Class Handles general exception in Brook.

EBrookHttp404 Class Handles the exception for 404 error.

EBrookHttp500 Class Handles the exception for 500 error.

### 7.4 Classes, Interfaces, Objects and Records

EBrook Class

---

Hierarchy

EBrook > Exception

## Description

Handles general exception in Brook.

## Methods

### Create

**Declaration** `public constructor Create(AInstance: TObject; const AMsg: string); overload;`

**Description** Creates an instance of EBrook.

### Create

**Declaration** `public constructor Create(AClass: TClass; const AMsg: string); overload;`

**Description** Creates an instance of EBrook.

### Create

**Declaration** `public constructor Create(AName: string; const AMsg: string); overload;`

**Description** Creates an instance of EBrook.

### CreateFmt

**Declaration** `public constructor CreateFmt(AInstance: TObject; const AMsg: string; const AArgs: array of const); overload;`

**Description** Creates an instance of EBrook with a formatted message.

### CreateFmt

**Declaration** `public constructor CreateFmt(AClass: TClass; const AMsg: string; const AArgs: array of const); overload;`

**Description** Creates an instance of EBrook with a formatted message.

### CreateFmt

**Declaration** `public constructor CreateFmt(AName: string; const AMsg: string; const AArgs: array of const); overload;`

**Description** Creates an instance of EBrook with a formatted message.

## EBrookHttp404 Class ---

## Hierarchy

EBrookHttp404 > EBrook(7.4) > Exception

## Description

Handles the exception for 404 error.

## EBrookHttp500 Class

---

### Hierarchy

EBrookHttp500 > EBrook(7.4) > Exception

## Description

Handles the exception for 500 error.

## 7.5 Types

### EBrookClass

---

**Declaration** `EBrookClass = class of EBrook;`

**Description** Is a metaclass for EBrook(7.4) class.

# Chapter 8

## Unit BrookHttpClient

### 8.1 Description

Brook framework, HTTP Client Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 8.2 Uses

- BrookClasses(3)
- BrookException(7)
- BrookMessages(12)
- BrookConsts(6)
- BrookHttpConsts(9)
- Classes
- SysUtils
- FGL

### 8.3 Overview

`E BrookHttpDef` Class Handles exceptions for `T BrookHttpDef`(8.4).

`E BrookHttpDefs` Class Handles exceptions for `T BrookHttpDefs`(8.4).

`E BrookHttpClient` Class Handles exceptions for `E BrookHttpClient`(8.4).

`TBrookHttpRequest` record Information returned after a request.

`TBrookHttpDef` Class Offers general abstract features for HTTP handling.

`TBrookHttpDefs` Class Registers HTTP definitions.

`TBrookHttpClient` Class Client to perform HTTP requests.

## 8.4 Classes, Interfaces, Objects and Records

### `EBrookHttpDef` Class ---

#### Hierarchy

`EBrookHttpDef` > `EBrook(7.4)` > Exception

#### Description

Handles exceptions for `TBrookHttpDef(8.4)`.

### `EBrookHttpDefs` Class ---

#### Hierarchy

`EBrookHttpDefs` > `EBrook(7.4)` > Exception

#### Description

Handles exceptions for `TBrookHttpDefs(8.4)`.

### `EBrookHttpClient` Class ---

#### Hierarchy

`EBrookHttpClient` > `EBrook(7.4)` > Exception

#### Description

Handles exceptions for `EBrookHttpClient(8.4)`.

### `TBrookHttpRequest` record ---

#### Description

Information returned after a request.



## Fields

**StatusCode**    `StatusCode: Integer;`  
**ReasonPhrase** `ReasonPhrase: string;`  
**Header**        `Header: string;`  
**Content**       `Content: string;`

## TBrookHttpDef Class

---

### Hierarchy

TBrookHttpDef > TBrookObject(3.4) > TObject

### Description

Offers general abstract features for HTTP handling.

### Properties

**ContentType**    `public property ContentType: string read GetContentType write SetContentType;`  
Content type of sending data.

**Contents**        `public property Contents: TStrings read GetContents;`  
Strings received from the request.

**Cookies**         `public property Cookies: TStrings read GetCookies;`  
Set cookies to be sent and/or received from the server.

**Document**       `public property Document: TStream read GetDocument;`  
Document received from the request.

**Headers**         `public property Headers: TStrings read GetHeaders;`  
Request headers.

**StatusCode**    `public property StatusCode: Integer read GetStatusCode;`  
Result code after successful request.

**ReasonPhrase** `public property ReasonPhrase: string read GetReasonPhrase;`  
Result text after successful request.

**Method**         `public property Method: string read GetMethod write SetMethod;`  
Method for requests.

**Url**             `public property Url: string read GetUrl write SetUrl;`  
URL that request is driven to.

**Client**            `public property Client: TObject read GetClient;`  
Is the instance of the HTTP client broker.

## Methods

### Create

**Declaration** `public constructor Create; virtual; abstract;`

**Description** Creates an instance of a `TBrookHttpDef(8.4)` class.

### Register

**Declaration** `public class procedure Register;`

**Description** Register the broker class.

### Unregister

**Declaration** `public class procedure Unregister;`

**Description** Unregister the broker class.

### GetLibrary

**Declaration** `public class function GetLibrary: string; virtual; abstract;`

**Description** Get the broker library name, for example: FCLWeb, Synapse, LNet, Indy etc.

### AddHeader

**Declaration** `public procedure AddHeader(const AName, AValue: string); virtual; abstract;`

**Description** Adds header, replacing an existing one if it exists.

### Request

**Declaration** `public function Request: Boolean; virtual; abstract;`

**Description** Sends request to server.

### Get

**Declaration** `public class function Get(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a GET HTTP request method.

## Post

**Declaration** `public class function Post(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a POST HTTP request method.

## Put

**Declaration** `public class function Put(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a PUT HTTP request method.

## Delete

**Declaration** `public class function Delete(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a DELETE HTTP request method.

## Options

**Declaration** `public class function Options(const AUrl: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by an OPTIONS HTTP request method.

## Head

**Declaration** `public class function Head(const AUrl: string; AHeaders: TStrings): Boolean; virtual; abstract;`

**Description** Sends request by a HEAD HTTP request method.

## PostForm

**Declaration** `public class function PostForm(const AUrl: string; AFormData, AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a POST HTTP request method, passing a form-data as parameter.

## PostForm

**Declaration** `public class function PostForm(const AUrl, AFormData: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a POST HTTP request method, passing a form-data as parameter.

## PostFile

**Declaration** `public class function PostFile(const AUrl, AFieldName, AFileName: string; AFile, AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a POST HTTP request method, passing a file as parameter.

## PostFile

**Declaration** `public class function PostFile(const AUrl, AFieldName, AFileName: string; AResponse: TStream): Boolean; virtual; abstract;`

**Description** Sends request by a POST HTTP request method, passing a file as parameter.

## TBrookHttpDefs Class

---

### Hierarchy

TBrookHttpDefs > TFPGList

### Description

Registers HTTP definitions.

### Methods

#### Service

**Declaration** `public class function Service: TBrookHttpDefs;`

**Description** Registers the service provided by this class.

#### Find

**Declaration** `public function Find(const ALibrary: string): TBrookHttpDefClass;`

**Description** Finds a database item by its library name.

#### ItemByLibrary

**Declaration** `public function ItemByLibrary(const ALibrary: string): TBrookHttpDefClass;`

**Description** Returns a database item by its library name.

## TBrookHttpClient Class

---

### Hierarchy

TBrookHttpClient > TBrookObject(3.4) > TObject

## Description

Client to perform HTTP requests.

## Methods

### Create

**Declaration** `public constructor Create(const ALibrary: string);`

**Description** Creates an instance of a `TBrookHttpClient(8.4)` class.

### SetLibrary

**Declaration** `public class procedure SetLibrary(const ALibrary: string);`

**Description** Specifies the library to be used by this class.

### Prepare

**Declaration** `public class procedure Prepare(out AHttp: TBrookHttpDef);`

**Description** Prepares an instance of `TBrookHttpDef`.

### Get

**Declaration** `public class function Get(const AUrl: string; AResponse: TStream): Boolean;`

**Description** Sends request by a GET HTTP request method.

### Post

**Declaration** `public class function Post(const AUrl: string; AResponse: TStream): Boolean;`

**Description** Sends request by a POST HTTP request method.

### Put

**Declaration** `public class function Put(const AUrl: string; AResponse: TStream): Boolean;`

**Description** Sends request by a PUT HTTP request method.

### Delete

**Declaration** `public class function Delete(const AUrl: string; AResponse: TStream): Boolean;`

**Description** Sends request by a DELETE HTTP request method.

## Options

**Declaration** `public class function Options(const AUrl: string; AResponse: TStream): Boolean;`

**Description** Sends request by an OPTIONS HTTP request method.

## Head

**Declaration** `public class function Head(const AUrl: string; AHeaders: TStrings): Boolean;`

**Description** Sends request by a HEAD HTTP request method.

## PostForm

**Declaration** `public class function PostForm(const AUrl: string; AFormData, AResponse: TStream): Boolean;`

**Description** Sends request by a POST HTTP request method, passing a form-data as parameter.

## PostForm

**Declaration** `public class function PostForm(const AUrl, AFormData: string; AResponse: TStream): Boolean;`

**Description** Sends request by a POST HTTP request method, passing a form-data as parameter.

## PostFile

**Declaration** `public class function PostFile(const AUrl, AFieldName, AFileName: string; AFile, AResponse: TStream): Boolean;`

**Description** Sends request by a POST HTTP request method, passing a file as parameter.

## PostFile

**Declaration** `public class function PostFile(const AUrl, AFieldName, AFileName: string; AResponse: TStream): Boolean;`

**Description** Sends request by a POST HTTP request method, passing a file as parameter.

## Request

**Declaration** `public function Request(AHttp: TBrookHttpDef): TBrookHttpRequest;`

**Description** Performs the request, passing a TBrookHttpDef as parameter.

## Request

**Declaration** `public function Request(const AMethod, AUrl: string): TBrookHttpRequest;`

**Description** Performs the request, passing the method and URL as parameter.

## Request

**Declaration** `public function Request(const AUrl: string): TBrookHttpRequest;`

**Description** Performs the request, passing URL as parameter.

## 8.5 Types

### TBrookHttpDefClass

---

**Declaration** `TBrookHttpDefClass = class of TBrookHttpDef;`

**Description** Is a metaclass for TBrookHttpDef(8.4) class.

### TBrookHttpDefsClass

---

**Declaration** `TBrookHttpDefsClass = class of TBrookHttpDefs;`

**Description** Is a metaclass for TBrookHttpDefs(8.4) class.

### TBrookHttpClientClass

---

**Declaration** `TBrookHttpClientClass = class of TBrookHttpClient;`

**Description** Is a metaclass for TBrookHttpClient(8.4) class.

# Chapter 9

## Unit BrookHttpConsts

### 9.1 Description

Brook framework, HTTP Consts Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 9.2 Uses

- BrookConsts(6)

### 9.3 Constants

**BROOK\_HTTP\_STATUS\_CODE\_CONTINUE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_CONTINUE = 100;

**BROOK\_HTTP\_STATUS\_CODE\_SWITCHING\_PROTOCOLS** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_SWITCHING\_PROTOCOLS = 101;

**BROOK\_HTTP\_STATUS\_CODE\_OK** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_OK = 200;

**BROOK\_HTTP\_STATUS\_CODE\_CREATED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_CREATED = 201;



**BROOK\_HTTP\_STATUS\_CODE\_ACCEPTED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_ACCEPTED = 202;

**BROOK\_HTTP\_STATUS\_CODE\_NON\_AUTHORITATIVE\_INFORMATION** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NON\_AUTHORITATIVE\_INFORMATION = 203;

**BROOK\_HTTP\_STATUS\_CODE\_NO\_CONTENT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NO\_CONTENT = 204;

**BROOK\_HTTP\_STATUS\_CODE\_RESET\_CONTENT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_RESET\_CONTENT = 205;

**BROOK\_HTTP\_STATUS\_CODE\_PARTIAL\_CONTENT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_PARTIAL\_CONTENT = 206;

**BROOK\_HTTP\_STATUS\_CODE\_MULTIPLE\_CHOICES** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_MULTIPLE\_CHOICES = 300;

**BROOK\_HTTP\_STATUS\_CODE\_MOVED\_PERMANENTLY** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_MOVED\_PERMANENTLY = 301;

**BROOK\_HTTP\_STATUS\_CODE\_FOUND** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_FOUND = 302;

**BROOK\_HTTP\_STATUS\_CODE\_SEE\_OTHER** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_SEE\_OTHER = 303;

**BROOK\_HTTP\_STATUS\_CODE\_NOT\_MODIFIED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NOT\_MODIFIED = 304;

**BROOK\_HTTP\_STATUS\_CODE\_USE\_PROXY** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_USE\_PROXY = 305;

**BROOK\_HTTP\_STATUS\_CODE\_TEMPORARY\_REDIRECT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_TEMPORARY\_REDIRECT = 307;

**BROOK\_HTTP\_STATUS\_CODE\_BAD\_REQUEST** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_BAD\_REQUEST = 400;

**BROOK\_HTTP\_STATUS\_CODE\_UNAUTHORIZED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_UNAUTHORIZED = 401;

**BROOK\_HTTP\_STATUS\_CODE\_PAYMENT\_REQUIRED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_PAYMENT\_REQUIRED = 402;

**BROOK\_HTTP\_STATUS\_CODE\_FORBIDDEN** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_FORBIDDEN = 403;

**BROOK\_HTTP\_STATUS\_CODE\_NOT\_FOUND** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NOT\_FOUND = 404;

**BROOK\_HTTP\_STATUS\_CODE\_METHOD\_NOT\_ALLOWED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_METHOD\_NOT\_ALLOWED = 405;

**BROOK\_HTTP\_STATUS\_CODE\_NOT\_ACCEPTABLE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NOT\_ACCEPTABLE = 406;

**BROOK\_HTTP\_STATUS\_CODE\_PROXY\_AUTHENTICATION\_REQUIRED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_PROXY\_AUTHENTICATION\_REQUIRED = 407;

**BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_TIMEOUT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_TIMEOUT = 408;

**BROOK\_HTTP\_STATUS\_CODE\_CONFLICT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_CONFLICT = 409;

**BROOK\_HTTP\_STATUS\_CODE\_GONE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_GONE = 410;

**BROOK\_HTTP\_STATUS\_CODE\_LENGTH\_REQUIRED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_LENGTH\_REQUIRED = 411;

**BROOK\_HTTP\_STATUS\_CODE\_PRECONDITION\_FAILED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_PRECONDITION\_FAILED = 412;

**BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_ENTITY\_TOO\_LARGE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_ENTITY\_TOO\_LARGE = 413;

**BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_URI\_TOO\_LONG** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_REQUEST\_URI\_TOO\_LONG = 414;

**BROOK\_HTTP\_STATUS\_CODE\_UNSUPPORTED\_MEDIA\_TYPE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_UNSUPPORTED\_MEDIA\_TYPE = 415;

**BROOK\_HTTP\_STATUS\_CODE\_REQUESTED\_RANGE\_NOT\_SATISFIABLE**

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_REQUESTED\_RANGE\_NOT\_SATISFIABLE = 416;

**BROOK\_HTTP\_STATUS\_CODE\_EXPECTATION\_FAILED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_EXPECTATION\_FAILED = 417;

**BROOK\_HTTP\_STATUS\_CODE\_INTERNAL\_SERVER\_ERROR** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_INTERNAL\_SERVER\_ERROR = 500;

**BROOK\_HTTP\_STATUS\_CODE\_NOT\_IMPLEMENTED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_NOT\_IMPLEMENTED = 501;

**BROOK\_HTTP\_STATUS\_CODE\_BAD\_GATEWAY** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_BAD\_GATEWAY = 502;

**BROOK\_HTTP\_STATUS\_CODE\_SERVICE\_UNAVAILABLE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_SERVICE\_UNAVAILABLE = 503;

**BROOK\_HTTP\_STATUS\_CODE\_GATEWAY\_TIMEOUT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_GATEWAY\_TIMEOUT = 504;

**BROOK\_HTTP\_STATUS\_CODE\_HTTP\_VERSION\_NOT\_SUPPORTED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_STATUS\_CODE\_HTTP\_VERSION\_NOT\_SUPPORTED = 505;

**BROOK\_HTTP\_REASON\_PHRASE\_CONTINUE** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_CONTINUE = 'Continue';

**BROOK\_HTTP\_REASON\_PHRASE\_SWITCHING\_PROTOCOLS** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_SWITCHING\_PROTOCOLS = 'Switching Protocols';

**BROOK\_HTTP\_REASON\_PHRASE\_OK** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_OK = 'OK';

**BROOK\_HTTP\_REASON\_PHRASE\_CREATED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_CREATED = 'Created';

**BROOK\_HTTP\_REASON\_PHRASE\_ACCEPTED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_ACCEPTED = 'Accepted';

**BROOK\_HTTP\_REASON\_PHRASE\_NON\_AUTHORITATIVE\_INFORMATION**

Declaration BROOK\_HTTP\_REASON\_PHRASE\_NON\_AUTHORITATIVE\_INFORMATION = 'Non-Authoritative Information';

**BROOK\_HTTP\_REASON\_PHRASE\_NO\_CONTENT** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_NO\_CONTENT = 'No Content';

**BROOK\_HTTP\_REASON\_PHRASE\_RESET\_CONTENT** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_RESET\_CONTENT = 'Reset Content';

**BROOK\_HTTP\_REASON\_PHRASE\_PARTIAL\_CONTENT** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_PARTIAL\_CONTENT = 'Partial Content';

**BROOK\_HTTP\_REASON\_PHRASE\_MULTIPLE\_CHOICES** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_MULTIPLE\_CHOICES = 'Multiple Choices';

**BROOK\_HTTP\_REASON\_PHRASE\_MOVED\_PERMANENTLY** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_MOVED\_PERMANENTLY = 'Moved Permanently';

**BROOK\_HTTP\_REASON\_PHRASE\_FOUND** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_FOUND = 'Found';

**BROOK\_HTTP\_REASON\_PHRASE\_SEE\_OTHER** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_SEE\_OTHER = 'See Other';

**BROOK\_HTTP\_REASON\_PHRASE\_NOT\_MODIFIED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_NOT\_MODIFIED = 'Not Modified';

**BROOK\_HTTP\_REASON\_PHRASE\_USE\_PROXY** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_USE\_PROXY = 'Use Proxy';

**BROOK\_HTTP\_REASON\_PHRASE\_TEMPORARY\_REDIRECT** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_TEMPORARY\_REDIRECT = 'Temporary Redirect';

**BROOK\_HTTP\_REASON\_PHRASE\_BAD\_REQUEST** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_BAD\_REQUEST = 'Bad Request';

**BROOK\_HTTP\_REASON\_PHRASE\_UNAUTHORIZED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_UNAUTHORIZED = 'Unauthorized';

**BROOK\_HTTP\_REASON\_PHRASE\_PAYMENT\_REQUIRED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_PAYMENT\_REQUIRED = 'Payment Required';

**BROOK\_HTTP\_REASON\_PHRASE\_FORBIDDEN** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_FORBIDDEN = 'Forbidden';

**BROOK\_HTTP\_REASON\_PHRASE\_NOT\_FOUND** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_NOT\_FOUND = 'Not Found';

**BROOK\_HTTP\_REASON\_PHRASE\_METHOD\_NOT\_ALLOWED** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_METHOD\_NOT\_ALLOWED = 'Method Not Allowed';

**BROOK\_HTTP\_REASON\_PHRASE\_NOT\_ACCEPTABLE** \_\_\_\_\_

Declaration BROOK\_HTTP\_REASON\_PHRASE\_NOT\_ACCEPTABLE = 'Not Acceptable';

**BROOK\_HTTP\_REASON\_PHRASE\_PROXY\_AUTHENTICATION\_REQUIRED**

Declaration BROOK\_HTTP\_REASON\_PHRASE\_PROXY\_AUTHENTICATION\_REQUIRED = 'Proxy Authentication Required';

**BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_TIMEOUT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_TIMEOUT = 'Request Timeout';

**BROOK\_HTTP\_REASON\_PHRASE\_CONFLICT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_CONFLICT = 'Conflict';

**BROOK\_HTTP\_REASON\_PHRASE\_GONE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_GONE = 'Gone';

**BROOK\_HTTP\_REASON\_PHRASE\_LENGTH\_REQUIRED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_LENGTH\_REQUIRED = 'Length Required';

**BROOK\_HTTP\_REASON\_PHRASE\_PRECONDITION\_FAILED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_PRECONDITION\_FAILED = 'Precondition Failed';

**BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_ENTITY\_TOO\_LARGE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_ENTITY\_TOO\_LARGE = 'Request Entity Too Large';

**BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_URI\_TOO\_LONG** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_REQUEST\_URI\_TOO\_LONG = 'Request-URI Too Long';

**BROOK\_HTTP\_REASON\_PHRASE\_UNSUPPORTED\_MEDIA\_TYPE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_UNSUPPORTED\_MEDIA\_TYPE = 'Unsupported Media Type';

**BROOK\_HTTP\_REASON\_PHRASE\_REQUESTED\_RANGE\_NOT\_SATISFIABLE**

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_REQUESTED\_RANGE\_NOT\_SATISFIABLE = 'Requested Range Not Satisfiable';

**BROOK\_HTTP\_REASON\_PHRASE\_EXPECTATION\_FAILED** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_EXPECTATION\_FAILED = 'Expectation Failed';

**BROOK\_HTTP\_REASON\_PHRASE\_INTERNAL\_SERVER\_ERROR** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REASON\_PHRASE\_INTERNAL\_SERVER\_ERROR = 'Internal Server Error';

**BROOK\_HTTP\_REASON\_PHRASE\_NOT\_IMPLEMENTED** \_\_\_\_\_

Declaration `BROOK_HTTP_REASON_PHRASE_NOT_IMPLEMENTED = 'Not Implemented';`

**BROOK\_HTTP\_REASON\_PHRASE\_BAD\_GATEWAY** \_\_\_\_\_

Declaration `BROOK_HTTP_REASON_PHRASE_BAD_GATEWAY = 'Bad Gateway';`

**BROOK\_HTTP\_REASON\_PHRASE\_SERVICE\_UNAVAILABLE** \_\_\_\_\_

Declaration `BROOK_HTTP_REASON_PHRASE_SERVICE_UNAVAILABLE = 'Service Unavailable';`

**BROOK\_HTTP\_REASON\_PHRASE\_GATEWAY\_TIMEOUT** \_\_\_\_\_

Declaration `BROOK_HTTP_REASON_PHRASE_GATEWAY_TIMEOUT = 'Gateway Timeout';`

**BROOK\_HTTP\_REASON\_PHRASE\_HTTP\_VERSION\_NOT\_SUPPORTED** \_\_\_\_\_

Declaration `BROOK_HTTP_REASON_PHRASE_HTTP_VERSION_NOT_SUPPORTED = 'HTTP Version Not Supported';`

**BROOK\_HTTP\_REQUEST\_METHOD\_GET** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_GET = 'GET';`

**BROOK\_HTTP\_REQUEST\_METHOD\_POST** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_POST = 'POST';`

**BROOK\_HTTP\_REQUEST\_METHOD\_PUT** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_PUT = 'PUT';`

**BROOK\_HTTP\_REQUEST\_METHOD\_DELETE** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_DELETE = 'DELETE';`

**BROOK\_HTTP\_REQUEST\_METHOD\_HEAD** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_HEAD = 'HEAD';`

**BROOK\_HTTP\_REQUEST\_METHOD\_OPTIONS** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_OPTIONS = 'OPTIONS';`

**BROOK\_HTTP\_REQUEST\_METHOD\_PATCH** \_\_\_\_\_

Declaration `BROOK_HTTP_REQUEST_METHOD_PATCH = 'PATCH';`

## **BROOK\_HTTP\_REQUEST\_METHOD\_PROPFIND** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_PROPFIND = 'PROPFIND';

## **BROOK\_HTTP\_REQUEST\_METHOD\_PROPPATCH** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_PROPPATCH = 'PROPPATCH';

## **BROOK\_HTTP\_REQUEST\_METHOD\_COPY** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_COPY = 'COPY';

## **BROOK\_HTTP\_REQUEST\_METHOD\_MOVE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_MOVE = 'MOVE';

## **BROOK\_HTTP\_REQUEST\_METHOD\_MKCOL** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_MKCOL = 'MKCOL';

## **BROOK\_HTTP\_REQUEST\_METHOD\_TRACE** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_TRACE = 'TRACE';

## **BROOK\_HTTP\_REQUEST\_METHOD\_LOCK** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_LOCK = 'LOCK';

## **BROOK\_HTTP\_REQUEST\_METHOD\_UNLOCK** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_REQUEST\_METHOD\_UNLOCK = 'UNLOCK';

## **BROOK\_HTTP\_HEADER\_CACHE\_CONTROL** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_CACHE\_CONTROL = 'Cache-Control';

## **BROOK\_HTTP\_HEADER\_CONTENT\_DESCRIPTION** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_CONTENT\_DESCRIPTION = 'Content-Description';

## **BROOK\_HTTP\_HEADER\_CONTENT\_DISPOSITION** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_CONTENT\_DISPOSITION = 'Content-Disposition';

## **BROOK\_HTTP\_HEADER\_ETAG** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_ETAG = 'ETag';



## **BROOK\_HTTP\_HEADER\_STATUS** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_STATUS = 'Status';

## **BROOK\_HTTP\_HEADER\_TRANSFER\_ENCODING** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_TRANSFER\_ENCODING = 'Transfer-Encoding';

## **BROOK\_HTTP\_HEADER\_X\_POWERED\_BY** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_X\_POWERED\_BY = 'X-Powered-By';

## **BROOK\_HTTP\_HEADER\_CHARSET** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_HEADER\_CHARSET = '; charset=';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_PLAIN** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_PLAIN = 'text/plain';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_HTML** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_HTML = 'text/html';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_CSS** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_CSS = 'text/css';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_XML** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_XML = 'text/xml';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_CVS** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_CVS = 'text/cvs';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_RICHTEXT** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_RICHTEXT = 'text/richtext';

## **BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_RTF** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_RTF = 'text/rtf';

## **BROOK\_HTTP\_CONTENT\_TYPE\_MULTIPART\_FORM\_DATA** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_MULTIPART\_FORM\_DATA = 'multipart/form-data';

## **BROOK\_HTTP\_CONTENT\_TYPE\_MULTIPART\_FORM\_DATA\_BOUNDARY**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_MULTIPART\_FORM\_DATA\_BOUNDARY =  
BROOK\_HTTP\_CONTENT\_TYPE\_MULTIPART\_FORM\_DATA + '; boundary=';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_JSON**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_JSON = 'application/json';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_XML**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_XML = 'application/xml';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_RDF\_XML**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_RDF\_XML = 'application/rdf+xml';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_X\_WWW\_FORM\_URLENCODED**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_X\_WWW\_FORM\_URLENCODED =  
'application/x-www-form-urlencoded';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_JAVASCRIPT**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_JAVASCRIPT = 'application/javascript';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_OCTET\_STREAM**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_OCTET\_STREAM = 'application/octet-stream';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_PDF**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_PDF = 'application/pdf';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_ZIP**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_ZIP = 'application/zip';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MSWORD**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MSWORD = 'application/msword';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MS\_EXCEL**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MS\_EXCEL = 'application/vnd.ms-excel';

## **BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MS\_POWERPOINT**

**Declaration** BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MS\_POWERPOINT = 'application/vnd.ms-powerpoint';

**BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MP4** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MP4 = 'application/mp4';

**BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MPEG4\_GENERIC** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_APP\_MPEG4\_GENERIC = 'application/mpeg4-generic';

**BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_PNG** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_PNG = 'image/png';

**BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_JPEG** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_JPEG = 'image/jpeg';

**BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_GIF** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_GIF = 'image/gif';

**BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_TIFF** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_IMAGE\_TIFF = 'image/tiff';

**BROOK\_HTTP\_CONTENT\_TYPE\_AUDIO\_MPEG** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_AUDIO\_MPEG = 'audio/mpeg';

**BROOK\_HTTP\_CONTENT\_TYPE\_VIDEO\_MPEG** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_VIDEO\_MPEG = 'video/mpeg';

**BROOK\_HTTP\_CONTENT\_TYPE\_VIDEO\_QUICKTIME** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_TYPE\_VIDEO\_QUICKTIME = 'video/quicktime';

**BROOK\_HTTP\_CACHE\_CONTROL\_NO\_CACHE** \_\_\_\_\_

Declaration BROOK\_HTTP\_CACHE\_CONTROL\_NO\_CACHE = 'no-cache';

**BROOK\_HTTP\_CACHE\_CONTROL\_NO\_STORE** \_\_\_\_\_

Declaration BROOK\_HTTP\_CACHE\_CONTROL\_NO\_STORE = 'no-store';

**BROOK\_HTTP\_CACHE\_CONTROL\_MAX\_AGE** \_\_\_\_\_

Declaration BROOK\_HTTP\_CACHE\_CONTROL\_MAX\_AGE = 'max-age=';

**BROOK\_HTTP\_CACHE\_CONTROL\_MAX\_STALE** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_MAX_STALE = 'max-stale';`

**BROOK\_HTTP\_CACHE\_CONTROL\_MIN\_FRESH** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_MIN_FRESH = 'min-fresh';`

**BROOK\_HTTP\_CACHE\_CONTROL\_NO\_TRANSFORM** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_NO_TRANSFORM = 'no-transform';`

**BROOK\_HTTP\_CACHE\_CONTROL\_ONLY\_IF\_CACHED** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_ONLY_IF_CACHED = 'only-if-cached';`

**BROOK\_HTTP\_CACHE\_CONTROL\_CACHE\_EXTENSION** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_CACHE_EXTENSION = 'cache-extension';`

**BROOK\_HTTP\_CACHE\_CONTROL\_PUBLIC** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_PUBLIC = 'public';`

**BROOK\_HTTP\_CACHE\_CONTROL\_PRIVATE** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_PRIVATE = 'private';`

**BROOK\_HTTP\_CACHE\_CONTROL\_MUST\_REVALIDATE** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_MUST_REVALIDATE = 'must-revalidate';`

**BROOK\_HTTP\_CACHE\_CONTROL\_PROXY\_REVALIDATE** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_PROXY_REVALIDATE = 'proxy-revalidate';`

**BROOK\_HTTP\_CACHE\_CONTROL\_S\_MAXAGE** \_\_\_\_\_

Declaration `BROOK_HTTP_CACHE_CONTROL_S_MAXAGE = 's-maxage';`

**BROOK\_HTTP\_CONTENT\_DISPOSITION\_INLINE** \_\_\_\_\_

Declaration `BROOK_HTTP_CONTENT_DISPOSITION_INLINE = 'inline';`

**BROOK\_HTTP\_CONTENT\_DISPOSITION\_ATTACHMENT** \_\_\_\_\_

Declaration `BROOK_HTTP_CONTENT_DISPOSITION_ATTACHMENT = 'attachment';`

**BROOK\_HTTP\_CONTENT\_ENCODING\_DEFLATE** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_ENCODING\_DEFLATE = 'deflate';

**BROOK\_HTTP\_CONTENT\_ENCODING\_GZIP** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_ENCODING\_GZIP = 'gzip';

**BROOK\_HTTP\_CONTENT\_ENCODING\_SDCH** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_ENCODING\_SDCH = 'sdch';

**BROOK\_HTTP\_CONTENT\_ENCODING\_X\_GZIP** \_\_\_\_\_

Declaration BROOK\_HTTP\_CONTENT\_ENCODING\_X\_GZIP = 'x-gzip';

**BROOK\_HTTP\_TRANSFER\_ENCODING\_CHUNKED** \_\_\_\_\_

Declaration BROOK\_HTTP\_TRANSFER\_ENCODING\_CHUNKED = 'chunked';

**BROOK\_HTTP\_CHARSET\_ISO\_8859\_1** \_\_\_\_\_

Declaration BROOK\_HTTP\_CHARSET\_ISO\_8859\_1 = 'iso-8859-1';

**BROOK\_HTTP\_CHARSET\_UTF\_8** \_\_\_\_\_

Declaration BROOK\_HTTP\_CHARSET\_UTF\_8 = 'utf-8';

**BROOK\_SRV\_ENV\_DOCUMENT\_ROOT** \_\_\_\_\_

Declaration BROOK\_SRV\_ENV\_DOCUMENT\_ROOT = 'DOCUMENT\_ROOT';

**BROOK\_SRV\_ENV\_SERVER\_ADDR** \_\_\_\_\_

Declaration BROOK\_SRV\_ENV\_SERVER\_ADDR = 'SERVER\_ADDR';

**BROOK\_SRV\_ENV\_SERVER\_ADMIN** \_\_\_\_\_

Declaration BROOK\_SRV\_ENV\_SERVER\_ADMIN = 'SERVER\_ADMIN';

**BROOK\_SRV\_ENV\_SERVER\_SOFTWARE** \_\_\_\_\_

Declaration BROOK\_SRV\_ENV\_SERVER\_SOFTWARE = 'SERVER\_SOFTWARE';

**BROOK\_SRV\_ENV\_SERVER\_NAME** \_\_\_\_\_

Declaration BROOK\_SRV\_ENV\_SERVER\_NAME = 'SERVER\_NAME';

## **BROOK\_SRV\_ENV\_GATEWAY\_INTERFACE** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_GATEWAY\_INTERFACE = 'GATEWAY\_INTERFACE';

## **BROOK\_SRV\_ENV\_SERVER\_PROTOCOL** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_SERVER\_PROTOCOL = 'SERVER\_PROTOCOL';

## **BROOK\_SRV\_ENV\_SERVER\_PORT** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_SERVER\_PORT = 'SERVER\_PORT';

## **BROOK\_SRV\_ENV\_REQUEST\_METHOD** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REQUEST\_METHOD = 'REQUEST\_METHOD';

## **BROOK\_SRV\_ENV\_REQUEST\_URI** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REQUEST\_URI = 'REQUEST\_URI';

## **BROOK\_SRV\_ENV\_PATH\_INFO** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_PATH\_INFO = 'PATH\_INFO';

## **BROOK\_SRV\_ENV\_PATH\_TRANSLATED** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_PATH\_TRANSLATED = 'PATH\_TRANSLATED';

## **BROOK\_SRV\_ENV\_SCRIPT\_FILENAME** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_SCRIPT\_FILENAME = 'SCRIPT\_FILENAME';

## **BROOK\_SRV\_ENV\_SCRIPT\_NAME** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_SCRIPT\_NAME = 'SCRIPT\_NAME';

## **BROOK\_SRV\_ENV\_QUERY\_STRING** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_QUERY\_STRING = 'QUERY\_STRING';

## **BROOK\_SRV\_ENV\_REMOTE\_HOST** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REMOTE\_HOST = 'REMOTE\_HOST';

## **BROOK\_SRV\_ENV\_REMOTE\_ADDR** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REMOTE\_ADDR = 'REMOTE\_ADDR';

## **BROOK\_SRV\_ENV\_REMOTE\_PORT** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REMOTE\_PORT = 'REMOTE\_PORT';

## **BROOK\_SRV\_ENV\_AUTH\_TYPE** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_AUTH\_TYPE = 'AUTH\_TYPE';

## **BROOK\_SRV\_ENV\_REMOTE\_USER** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REMOTE\_USER = 'REMOTE\_USER';

## **BROOK\_SRV\_ENV\_REMOTE\_IDENT** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_REMOTE\_IDENT = 'REMOTE\_IDENT';

## **BROOK\_SRV\_ENV\_CONTENT\_TYPE** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_CONTENT\_TYPE = 'CONTENT\_TYPE';

## **BROOK\_SRV\_ENV\_CONTENT\_LENGTH** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_CONTENT\_LENGTH = 'CONTENT\_LENGTH';

## **BROOK\_SRV\_ENV\_X\_FORWARDED\_FOR** \_\_\_\_\_

**Declaration** BROOK\_SRV\_ENV\_X\_FORWARDED\_FOR = 'X\_FORWARDED\_FOR';

## **BROOK\_CLT\_ENV\_HTTP\_COOKIE** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_COOKIE = 'HTTP\_COOKIE';

## **BROOK\_CLT\_ENV\_HTTP\_REFERER** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_REFERER = 'HTTP\_REFERER';

## **BROOK\_CLT\_ENV\_HTTP\_USER\_AGENT** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_USER\_AGENT = 'HTTP\_USER\_AGENT';

## **BROOK\_CLT\_ENV\_HTTP\_IF\_MODIFIED\_SINCE** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_IF\_MODIFIED\_SINCE = 'HTTP\_IF\_MODIFIED\_SINCE';

## **BROOK\_CLT\_ENV\_HTTP\_HOST** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_HOST = 'HTTP\_HOST';

## **BROOK\_CLT\_ENV\_HTTP\_CONNECTION** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_CONNECTION = 'HTTP\_CONNECTION';

## **BROOK\_CLT\_ENV\_HTTP\_CACHE\_CONTROL** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_CACHE\_CONTROL = 'HTTP\_CACHE\_CONTROL';

## **BROOK\_CLT\_ENV\_HTTP\_ACCEPT** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_ACCEPT = 'HTTP\_ACCEPT';

## **BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_ENCODING** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_ENCODING = 'HTTP\_ACCEPT\_ENCODING';

## **BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_LANGUAGE** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_LANGUAGE = 'HTTP\_ACCEPT\_LANGUAGE';

## **BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_CHARSET** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_ACCEPT\_CHARSET = 'HTTP\_ACCEPT\_CHARSET';

## **BROOK\_CLT\_ENV\_HTTP\_IF\_NONE\_MATCH** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_IF\_NONE\_MATCH = 'HTTP\_IF\_NONE\_MATCH';

## **BROOK\_CLT\_ENV\_HTTP\_X\_REQUESTED\_WITH** \_\_\_\_\_

**Declaration** BROOK\_CLT\_ENV\_HTTP\_X\_REQUESTED\_WITH = 'HTTP\_X\_REQUESTED\_WITH';

## **BROOK\_HTTP\_RESPONSE\_TEMPLATE\_NOT\_FOUND** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_RESPONSE\_TEMPLATE\_NOT\_FOUND = '<html><head><title>Page not found</title></head><body>' + '<h1>404 - Page not found</h1></body></html>';

## **BROOK\_HTTP\_RESPONSE\_TEMPLATE\_INTERNAL\_SERVER\_ERROR** \_\_\_\_\_

**Declaration** BROOK\_HTTP\_RESPONSE\_TEMPLATE\_INTERNAL\_SERVER\_ERROR = '<html><head><title>Internal server error</title></head><body>' + '<h1>500 - Internal server error</h1>' + '<p style="color: red;" >Error: @error</p>' + '</body></html>';



## 9.4 Variables

**BROOK\_HTTP\_CLIENT\_DEFAULT\_LIBRARY** \_\_\_\_\_

Declaration `BROOK_HTTP_CLIENT_DEFAULT_LIBRARY: string = ES;`

# Chapter 10

## Unit BrookHttpDefs

### 10.1 Description

Brook framework, HttpDefs Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 10.2 Uses

- HttpDefs

### 10.3 Types

#### TBrookRequest

---

**Declaration** TBrookRequest = TRequest;

**Description** Alias to TRequest.

#### TBrookResponse

---

**Declaration** TBrookResponse = TResponse;

**Description** Alias to TResponse.

#### TBrookUploadedFile

---

**Declaration** TBrookUploadedFile = TUploadedFile;

**Description** Alias to TUploadedFile.

## **TBrookUploadedFiles**

---

**Declaration** `TBrookUploadedFiles = TUploadedFiles;`

**Description** Alias to TUploadedFiles.

# Chapter 11

## Unit BrookHttpUtils

### 11.1 Description

Brook framework, HTTP Utilities Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 11.2 Uses

- BrookHttpClient(8)
- BrookMessages(12)
- BrookHttpConsts(9)
- BrookConsts(6)
- BrookUtils(16)
- HttpDefs
- SysUtils

### 11.3 Overview

`BrookDateTimeToGMT` Converts TDateTime to a GMT format.

`BrookSameUrl` Compare two URLs ignoring a possible final slash.

`BrookStatusCodeToReasonPhrase` Returns the reason phrase corresponding to a status code.

`BrookReasonPhraseToStatusCode` Returns the status code corresponding to a reason phrase.

**BrookGetAcceptEncodingSet** Returns a set of HTTP AcceptEncoding header.

**BrookGetAcceptEncoding** Returns a string of HTTP AcceptEncoding.

**BrookMimeTypeFromFileExt** Returns a MIME type by file extension.

**BrookMimeTypeFromFileName** Returns a MIME type by file name.

**BrookFileExtFromMimeType** Returns a file extension by MIME type.

**BrookExtractUrlFileName** Extracts the file name of an URL.

**BrookExtractUrlFileName** Extracts the file name of an URL and escapes it.

**BrookRequestMethodToStr** Returns the string corresponding to a TBrookRequestMethod.

**BrookStrToRequestMethod** Returns the TBrookRequestMethod corresponding to a string.

**BrookHttpRequest** Perform HTTP requests.

## 11.4 Functions and Procedures

### BrookDateTimeToGMT

---

**Declaration** `function BrookDateTimeToGMT(const ADateTime: TDateTime): string;`

**Description** Converts TDateTime to a GMT format.

### BrookSameUrl

---

**Declaration** `function BrookSameUrl(AUrl1, AUrl2: string): Boolean;`

**Description** Compare two URLs ignoring a possible final slash.

### BrookStatusCodeToReasonPhrase

---

**Declaration** `function BrookStatusCodeToReasonPhrase(const AStatusCode: Word): string;`

**Description** Returns the reason phrase corresponding to a status code.

### BrookReasonPhraseToStatusCode

---

**Declaration** `function BrookReasonPhraseToStatusCode(const AReasonPhrase: string): Word;`

**Description** Returns the status code corresponding to a reason phrase.

### BrookGetAcceptEncodingSet

---

**Declaration** `function BrookGetAcceptEncodingSet(const AAcceptEncoding: ShortString): TBrookAcceptEncodingSet;`

**Description** Returns a set of HTTP AcceptEncoding header.

## BrookGetAcceptEncoding

---

**Declaration** `function BrookGetAcceptEncoding( const AAcceptEncoding: TBrookAcceptEncodingSet): string;`

**Description** Returns a string of HTTP AcceptEncoding.

## BrookMimeTypeFromFileExt

---

**Declaration** `function BrookMimeTypeFromFileExt(const AValue: string): string;`

**Description** Returns a MIME type by file extension.

## BrookMimeTypeFromFileName

---

**Declaration** `function BrookMimeTypeFromFileName(const AValue: string): string;`

**Description** Returns a MIME type by file name.

## BrookFileExtFromMimeType

---

**Declaration** `function BrookFileExtFromMimeType(const AValue: string): string;`

**Description** Returns a file extension by MIME type.

## BrookExtractUrlFileName

---

**Declaration** `function BrookExtractUrlFileName(const AUrl: string): string;`

**Description** Extracts the file name of an URL.

## BrookExtractUrlFileName

---

**Declaration** `function BrookExtractUrlFileName(const AUrl: string; const AEscapeQueryString: Boolean): string;`

**Description** Extracts the file name of an URL and escapes it.

## BrookRequestMethodToStr

---

**Declaration** `function BrookRequestMethodToStr(const AMethod: TBrookRequestMethod): string;`

**Description** Returns the string corresponding to a TBrookRequestMethod.

## BrookStrToRequestMethod

---

**Declaration** `function BrookStrToRequestMethod(const AMethod: string): TBrookRequestMethod;`

**Description** Returns the TBrookRequestMethod corresponding to a string.

## BrookHttpRequest

---

**Declaration** `function BrookHttpRequest(const AUrl: string; const AMethod:  
TBrookRequestMethod = rmGet; const AHttpClientLibrary: string = ES):  
TBrookHTTPResult;`

**Description** Perform HTTP requests. (allows all request methods)

## 11.5 Types

### TBrookAcceptEncodingSet

---

**Declaration** `TBrookAcceptEncodingSet = set of (aeDeflate, aeGzip, aeSdch, aeXGzip);`

**Description** Defines a set to represent the AcceptEncoding HTTP header.

# Chapter 12

## Unit BrookMessages

### 12.1 Description

Brook framework, Messages Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 12.2 Variables

#### **SBrookInvalidRequestMethodError** \_\_\_\_\_

**Declaration** SBrookInvalidRequestMethodError: string = 'Invalid request method: %s';

#### **SBrookItemNotFoundError** \_\_\_\_\_

**Declaration** SBrookItemNotFoundError: string = 'Item "%s" not found.';

#### **SBrookFileNotFoundError** \_\_\_\_\_

**Declaration** SBrookFileNotFoundError: string = 'File not found: %s';

#### **SBrookNoRequestMethodError** \_\_\_\_\_

**Declaration** SBrookNoRequestMethodError: string = 'No REQUEST.METHOD passed from server.';

#### **SBrookNoApplicationRegisteredError** \_\_\_\_\_

**Declaration** SBrookNoApplicationRegisteredError: string = 'No application registered.';



### **SBrookApplicationAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookApplicationAlreadyRegisteredError: string = 'The application is already registered.';

### **SBrookRegiterTBrookActionError** \_\_\_\_\_

**Declaration** SBrookRegiterTBrookActionError: string = 'Unable to register the type TBrookAction directly.';

### **SBrookActionAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookActionAlreadyRegisteredError: string = 'The action "%s" is already registered.';

### **SBrookDefaultActionAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookDefaultActionAlreadyRegisteredError: string = 'There is already a default action registered.';

### **SBrookPatternAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookPatternAlreadyRegisteredError: string = 'There is already an action registered with the pattern "%s".';

### **SBrookRouterServiceAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookRouterServiceAlreadyRegisteredError: string = 'The router service is already registered.';

### **SBrookNoRouterServiceRegisteredError** \_\_\_\_\_

**Declaration** SBrookNoRouterServiceRegisteredError: string = 'No router service registered.';

### **SBrookNoRouteRegisteredError** \_\_\_\_\_

**Declaration** SBrookNoRouteRegisteredError: string = 'No route registered.';

### **SBrookCfgFileNotFoundError** \_\_\_\_\_

**Declaration** SBrookCfgFileNotFoundError: string = 'The config file was not found: "%s".';

### **SBrookNotNilError** \_\_\_\_\_

**Declaration** SBrookNotNilError: string = '"%s" must not be nil.';

### **SBrookEmptyLibraryNameError** \_\_\_\_\_

**Declaration** SBrookEmptyLibraryNameError: string = 'The library name must not be empty.';

### **SBrookMethodNotAllowedError** \_\_\_\_\_

**Declaration** SBrookMethodNotAllowedError: string = 'HTTP method not allowed for the requested resource.';

### **SBrookConstraintAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookConstraintAlreadyRegisteredError: string = 'The constraint "%s" is already registered.';

### **SBrookConstraintsServiceAlreadyRegisteredError** \_\_\_\_\_

**Declaration** SBrookConstraintsServiceAlreadyRegisteredError: string = 'The constraints service is already registered.';

### **SBrookNoConstraintsServiceRegisteredError** \_\_\_\_\_

**Declaration** SBrookNoConstraintsServiceRegisteredError: string = 'No constraints service registered.';

# Chapter 13

## Unit BrookMiddleware

### 13.1 Description

Brook framework, Middleware Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 13.2 Uses

- BrookClasses(3)
- BrookAction(1)
- BrookRouter(14)
- BrookException(7)
- HTTPDefs

### 13.3 Overview

`EBrookMiddleware` Class Handles exceptions for `TBrookMiddleware`(13.4).

`TBrookMiddleware` Class Intermediates two classes through a `TBrookExecuteActionEvent` event.

## 13.4 Classes, Interfaces, Objects and Records

### EBrookMiddleware Class

---

#### Hierarchy

EBrookMiddleware > EBrook(7.4) > Exception

#### Description

Handles exceptions for TBrookMiddleware(13.4).

### TBrookMiddleware Class

---

#### Hierarchy

TBrookMiddleware > TBrookObject(3.4) > TObject

#### Description

Intermediates two classes through a TBrookExecuteActionEvent event.

#### Methods

##### Create

**Declaration** `public constructor Create(ABoundEvent: PBrookExecuteActionEvent); overload; virtual;`

**Description** Creates an instance of a TBrookMiddleware(13.4) class.

##### Execute

**Declaration** `public procedure Execute(ASender: TObject; AAction: TBrookAction; ARoute: TBrookRoute); virtual;`

**Description** Is triggered when the TBrookExecuteActionEvent event bound in this class is executed.

##### BindExecution

**Declaration** `public procedure BindExecution(AEvent: PBrookExecuteActionEvent);`

**Description** Binds a TBrookExecuteActionEvent event to this class keeping the implementation of a previously declared event.

## 13.5 Types

**TBrookMiddlewareClass** 

---

**Declaration** `TBrookMiddlewareClass = class of TBrookMiddleware;`

**Description** Is a metaclass for TBrookMiddleware(13.4) class.

# Chapter 14

## Unit BrookRouter

### 14.1 Description

Brook framework, Router Classes

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 14.2 Uses

- BrookClasses(3)
- BrookHttpDefs(10)
- BrookException(7)
- BrookAction(1)
- BrookUtils(16)
- BrookConsts(6)
- BrookMessages(12)
- BrookHTTPConsts(9)
- HTTPDefs
- Classes
- SysUtils
- StrUtils

## 14.3 Overview

**EBrookRoutes** Class Handles exceptions for **TBrookRoutes**(14.4).

**EBrookRouter** Class Handles exceptions for **TBrookRouter**(14.4).

**TBrookRoute** record Defines a route item.

**TBrookRoutes** Class Defines a list of routes.

**TBrookRouter** Class Provides features for the route handling.

## 14.4 Classes, Interfaces, Objects and Records

### **EBrookRoutes** Class

---

#### **Hierarchy**

**EBrookRoutes** > **EBrook**(7.4) > Exception

#### **Description**

Handles exceptions for **TBrookRoutes**(14.4).

### **EBrookRouter** Class

---

#### **Hierarchy**

**EBrookRouter** > **EBrook**(7.4) > Exception

#### **Description**

Handles exceptions for **TBrookRouter**(14.4).

### **TBrookRoute** record

---

#### **Description**

Defines a route item.

#### **Fields**

**ActionClass** ActionClass: **TBrookActionClass**;  
Specifies the class of the action to be called.

**Method** Method: **TBrookRequestMethod**;  
Specifies a HTTP request method of the action to be called.

**Pattern** Pattern: **string**;  
Specifies the patter of the action to be called.

## TBrookRoutes Class

---

### Hierarchy

TBrookRoutes > TBrookObject(3.4) > TObject

### Description

Defines a list of routes.

### Properties

**Items** public property Items[constAIndex:Integer]: PBrookRoute read GetItems write SetItems;

The list of routes.

### Methods

#### Create

**Declaration** public constructor Create;

**Description** Creates an instance of a TBrookRoutes(14.4) class.

#### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookRoutes(14.4) class.

#### Clear

**Declaration** public procedure Clear;

**Description** Clears all routes.

#### Count

**Declaration** public function Count: Integer;

**Description** Returns the number of registered routes.

#### Add

**Declaration** public function Add(AActionClass: TBrookActionClass; const APattern: string; const AMethod: TBrookRequestMethod; const ADefault: Boolean): Integer;

**Description** Adds a route item.



### GetDefaultActionClass

**Declaration** `public procedure GetDefaultActionClass(out AClass: TBrookActionClass; out AIndex: Integer);`

**Description** Get the default action class.

### PatternByActionClass

**Declaration** `public function PatternByActionClass(AClass: TBrookActionClass): string;`

**Description** Get the registered pattern of a class.

### ActionClassByPattern

**Declaration** `public function ActionClassByPattern(const APattern: string): TBrookActionClass;`

**Description** Get the action class from a patter.

### ActionClassByClassName

**Declaration** `public function ActionClassByClassName(const AName: string): TBrookActionClass;`

**Description** Get an action class from its class name.

## TBrookRouter Class

---

### Hierarchy

TBrookRouter > TBrookObject(3.4) > TObject

### Description

Provides features for the route handling.

### Properties

**Routes** `public property Routes: TBrookRoutes read FRoutes write FRoutes;`  
List of available routes.

**AfterExecuteAction** `public property AfterExecuteAction: TBrookExecuteActionEvent read FAfterExecuteAction write FAfterExecuteAction;`  
Is triggered after the router executes a action.

**AfterMatchPattern** `public property AfterMatchPattern: TBrookMatchPatternEvent read FAfterMatchPattern write FAfterMatchPattern;`  
Is triggered after the router matches a pattern.

**AfterRoute**            public property AfterRoute: TBrookRouteEvent read FAfterRoute  
                              write FAfterRoute;  
                              Is triggered after the router is routing.

**BeforeExecuteAction** public property BeforeExecuteAction: TBrookExecuteActionEvent read  
                              FBeforeExecuteAction write FBeforeExecuteAction;  
                              Is triggered before the router executes a action.

**BeforeMatchPattern** public property BeforeMatchPattern: TBrookMatchPatternEvent read  
                              FBeforeMatchPattern write FBeforeMatchPattern;  
                              Is triggered before the router matches a pattern.

**BeforeRoute**            public property BeforeRoute: TBrookRouteEvent read FBeforeRoute  
                              write FBeforeRoute;  
                              Is triggered before the router is routing.

## Methods

### Create

**Declaration** public constructor Create; virtual;

**Description** Creates an instance of a TBrookRouter(14.4) class.

### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookRouter(14.4) class.

### GetServiceClass

**Declaration** public class function GetServiceClass: TBrookRouterClass;

**Description** Return the service class provided by this class.

### RegisterService

**Declaration** public class procedure RegisterService;

**Description** Registers the service provided by this class.

### UnregisterService

**Declaration** public class procedure UnregisterService;

**Description** Unregisters the service provided by this class.

## Service

**Declaration** `public class function Service: TBrookRouter;`

**Description** Return an instance of this class.

## RootUrl

**Declaration** `public class function RootUrl: string;`

**Description** Return the root URL.

## RootUrl

**Declaration** `public class function RootUrl(ARequest: TBrookRequest): string;`

**Description** Return the root URL passing TBrookRequest as param.

## MethodNotAllowed

**Declaration** `public class procedure MethodNotAllowed(AResponse: TBrookResponse);`

**Description** Sends the HTTP "NotAllowed" status code to the response.

## UrlFor

**Declaration** `public function UrlFor(AActionClass: TBrookActionClass; const AParams: array of string): string; overload;`

**Description** Creates an URL for an action informing an array of parameters. Exemple:

```
procedure TMyAction.Get;
begin
    // When calling with http://localhost/cgi-bin/cgi1/foo/myvalue
    // the output will be /cgi-bin/cgi1/foo/myvalue
    Write(UrlFor(TMyAction, ['myvalue']));
end;

initialization
    TMyAction.Register('/foo/:myvar')
```

## UrlFor

**Declaration** `public function UrlFor(AClassName: string; const AParams: array of string): string; overload;`

**Description** Creates an URL for an action passing an array of parameters however informing the class name as string

## Canonicalize

**Declaration** `public function Canonicalize(ARequest: TBrookRequest; AResponse: TBrookResponse): Boolean;`

**Description** Adds an slash to the end of the URL if does not exist.

## MatchPattern

**Declaration** `public function MatchPattern(APattern, APathInfo: string; out ARedirect: Boolean; out ANames, AValues: TBrookArrayOfString): Boolean; virtual;`

**Description** Checks if the given parameters match with a registered route.

## Route

**Declaration** `public procedure Route(ARequest: TBrookRequest; AResponse: TBrookResponse); virtual;`

**Description** Runs the route processing.

# 14.5 Types

## TBrookRoutesClass

---

**Declaration** `TBrookRoutesClass = class of TBrookRoutes;`

**Description** Is a metaclass for TBrookRoutes(14.4) class.

## TBrookRouterClass

---

**Declaration** `TBrookRouterClass = class of TBrookRouter;`

**Description** Is a metaclass for TBrookRouter(14.4) class.

## PBrookRoute

---

**Declaration** `PBrookRoute = ^TBrookRoute;`

**Description** Defines a pointer to the route item.

## TBrookMatchPatternEvent

---

**Declaration** `TBrookMatchPatternEvent = procedure(ASender: TObject; APattern, APathInfo: string; out ARedirect: Boolean; out ANames, AValues: TBrookArrayOfString) of object;`

**Description** Is a type to \*MatchPattern event.

## **PBrookMatchPatternEvent**

---

**Declaration** PBrookMatchPatternEvent = ^TBrookMatchPatternEvent;

**Description** Defines a pointer to the match pattern event.

## **TBrookRouteEvent**

---

**Declaration** TBrookRouteEvent = procedure(ASender: TObject; ARequest: TBrookRequest; AResponse: TBrookResponse) of object;

**Description** Is a type to \*Route event.

## **PBrookRouteEvent**

---

**Declaration** PBrookRouteEvent = ^TBrookRouteEvent;

**Description** Defines a pointer to the route event.

## **TBrookExecuteActionEvent**

---

**Declaration** TBrookExecuteActionEvent = procedure(ASender: TObject; AAction: TBrookAction; ARequest: TBrookRequest; AResponse: TBrookResponse; ARoute: TBrookRoute; var AHandled: Boolean) of object;

**Description** Is a type to \*ExecuteAction event.

## **PBrookExecuteActionEvent**

---

**Declaration** PBrookExecuteActionEvent = ^TBrookExecuteActionEvent;

**Description** Defines a pointer to the execute action event.

# Chapter 15

## Unit BrookSession

### 15.1 Description

Brook framework, Session Class

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 15.2 Uses

- BrookClasses(3)
- BrookHttpDefs(10)
- BrookUtils(16)
- BrookException(7)
- BrookConsts(6)
- Classes
- SysUtils
- HTTPDefs
- DateUtils

### 15.3 Overview

EBrookSession Class Handles exceptions for TBrookSession(15.4).

TBrookSession Class Defines features to the section handling.

TBrookGSession Class Defines features to the section mapping field values to object.

## 15.4 Classes, Interfaces, Objects and Records

### EBrookSession Class

---

#### Hierarchy

EBrookSession > EBrook(7.4) > Exception

#### Description

Handles exceptions for TBrookSession(15.4).

### TBrookSession Class

---

#### Hierarchy

TBrookSession > TBrookComponent(3.4) > TComponent

#### Description

Defines features to the section handling.

#### Properties

**CookieName** public property CookieName: string read FCookieName write FCookieName;  
Set the session cookie name.

**CookieDomain** public property CookieDomain: string read FCookieDomain write  
FCookieDomain;  
Set the session cookie domain.

**CookiePath** public property CookiePath: string read FCookiePath write FCookiePath;  
Set the session cookie path.

**CookieSecure** public property CookieSecure: Boolean read FCookieSecure write  
FCookieSecure;  
Set the session cookie secure.

**CookieExpires** public property CookieExpires: TDateTime read FCookieExpires write  
FCookieExpires;  
Set the session cookie expiration.

**Fields** public property Fields: TStrings read FFields;  
The session fields.

**Directory**      public property Directory: string read FDirectory write FDirectory;  
Set the name of session directory.

**Expired**        public property Expired: Boolean read IsExpired;  
Returns True if the session has expired.

**Empty**          public property Empty: Boolean read IsEmpty;  
Returns True if the session fields is empty.

**SID**             public property SID: string read FSID write FSID;  
Get or set the session ID.

**Started**        public property Started: Boolean read IsStarted;  
Checks if the session has started.

**FileName**       public property FileName: TFileName read FFileName write FFileName;  
The session file name.

**FilePrefix**     public property FilePrefix: ShortString read FFilePrefix write  
FFilePrefix;  
The session file prefix.

**TimeOut**        public property TimeOut: Integer read FTimeOut write FTimeOut;  
The remaining seconds for the session finish.

**HttpOnly**       public property HttpOnly: Boolean read FHttpOnly write FHttpOnly;  
Informs if the session cookie is accessible only by HTTP requests, if True, the JavaScript  
access is not allowed.

## Methods

### Create

**Declaration** public constructor Create(AOwner: TComponent); override;

**Description** Creates an instance of a TBrookSession(15.4) class.

### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookSession(15.4) class.

### GetFields

**Declaration** public procedure GetFields(AObject: TObject);

**Description** Get an object with the fields coming from session.



### IsExpired

**Declaration** `public function IsExpired: Boolean; virtual;`

**Description** Returns True if the session has expired.

### IsEmpty

**Declaration** `public function IsEmpty: Boolean; virtual;`

**Description** Returns True if the session fields is empty.

### GenerateID

**Declaration** `public function GenerateID: string; virtual;`

**Description** Creates an ID for the session.

### Start

**Declaration** `public procedure Start(ARequest: TBrookRequest); virtual;`

**Description** Starts the session.

### Finish

**Declaration** `public procedure Finish(AResponse: TBrookResponse); virtual;`

**Description** Terminates the session.

### Expire

**Declaration** `public procedure Expire(ARequest: TBrookRequest; AResponse: TBrookResponse); virtual;`

**Description** Expires the session.

### Exists

**Declaration** `public function Exists(const AName: string): Boolean;`

**Description** Checks if a name exists in fields.

## TBrookGSession Class ---

### Hierarchy

TBrookGSession > TBrookSession(15.4) > TBrookComponent(3.4) > TComponent

## Description

Defines features to the section mapping field values to object.

## Properties

**Entity** public property Entity: T read FEntity write FEntity;  
Maps field values to object.

## Methods

### Create

**Declaration** public constructor Create(AOwner: TComponent); override;

**Description** Creates an instance of a TBrookGSession(15.4) class.

### Destroy

**Declaration** public destructor Destroy; override;

**Description** Frees an instance of TBrookGSession(15.4) class.

## 15.5 Types

### TBrookSessionClass

---

**Declaration** TBrookSessionClass = class of TBrookSession;

**Description** Is a metaclass for TBrookSession(15.4) class.

# Chapter 16

## Unit BrookUtils

### 16.1 Description

Brook framework, Utilities Unit

Copyright (C) 2014 Silvio Clecio

See the file LICENSE.txt, included in this distribution, for details about the copyright.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

### 16.2 Uses

- BrookException(7)
- BrookMessages(12)
- BrookConsts(6)
- BrookHTTPConsts(9)
- CustWeb
- Classes
- SysUtils
- TypInfo

### 16.3 Overview

TBrookSettings record Stores the Brook configurations.

BrookStartsChar Check whether a string starts with a given character.

BrookEndsChar Check whether a string ends with a given character.

**BrookExtractPathLevels** Get the next pathinfo level.

**BrookGetPathLevel** Get the path level passing the respective index.

**BrookGetPathLevels** Get the path from the level correspondent to the index to the last level.

**BrookMatchMethod** Checks if a string is equivalent an enumerator representing a HTTP request method.

**BrookFileDate** Get the datetime of a file.

**BrookDumpStack** Writes a backtrace of the current exception.

**BrookDumpStackTrace** Writes a stack trace of the current exception.

**BrookExcludeTrailingUrlDelimiter** Ensures Url ends without delimiter.

**BrookIncludeTrailingUrlDelimiter** Ensures Url ends with delimiter.

**BrookExists** Checks if a string exists in an array of strings.

**BrookExists** Checks (ignoring case) if a string exists in an array of strings.

**BrookStringToObject** Fills a published property of an object passing the property as **PPropInfo** and value as **string**.

**BrookStringToObject** Fills a published property of an object passing the name and value as **string**.

**BrookSafeStringToObject** Fills a published property of an object passing the name and value as **string** and checking the params.

**BrookStringsToObject** Fills the published properties of an object passing the names and values as a list of strings.

**BrookStringsToObject** Fills the published properties of an object passing the names and values as a list of strings.

**BrookStringsToObject** Fills the published properties of an object passing the names and values as a list of strings.

**BrookSafeStringsToObject** Fills the published properties of an object passing the names and values as a list of strings and checking the params.

**BrookSafeStringsToObject** Fills the published properties of an object passing the names and values as a list of strings and checking the params.

**BrookSafeStringsToObject** Fills the published properties of an object passing the names and values as a list of strings and checking the params.

**BrookObjectToString** Reads a published property of an object passing the property as **PPropInfo** and getting the value as **string**.

**BrookObjectToString** Reads a published property of an object passing the name as **string** and getting the value as **string**.

**BrookSafeObjectToString** Reads a published property of an object passing the name, getting the value as string and checking the params.

**BrookObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings.

**BrookObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings.

**BrookObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings.

**BrookSafeObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings and checking the params.

**BrookSafeObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings and checking the params.

**BrookSafeObjectToStrings** Reads the published properties of an object getting the names and values as a list of strings and checking the params.

**BrookCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookCopyObject** Copies the value of all properties from one object to another.

**BrookCopyObject** Copies the value of all properties from one object to another.

**BrookCopyObject** Copies the value of all properties from one object to another.

**BrookSafeCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookSafeCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookSafeCopyObject** Copies the value of all properties from one object to another passing the prop.

**BrookSafeCopyObject** Copies the value of all properties from one object to another and checking the params.

**BrookSafeCopyObject** Copies the value of all properties from one object to another and checking the params.

**BrookSafeCopyObject** Copies the value of all properties from one object to another and checking the params.

## 16.4 Classes, Interfaces, Objects and Records

**TBrookSettings** record \_\_\_\_\_

### Description

Stores the Brook configurations.

## Fields

<b>Mapped</b>	<b>Mapped:</b> Boolean; Enables the mapping of HTTP request methods.
<b>Charset</b>	<b>Charset:</b> ShortString; Set the default application Charset.
<b>ContentType</b>	<b>ContentType:</b> ShortString; Set the default application Content-Type.
<b>Page404</b>	<b>Page404:</b> string; Set the 404 HTML page. The string will be sent as is.
<b>Page404File</b>	<b>Page404File:</b> string; Set the 404 HTML page file. The file content will be sent. This has higher precedence than <code>TBrookSettings.Page404</code> so when both are set, this will be processed first and only if the file is not found or cannot be read the system will fallback to <code>TBrookSettings.Page404</code>
<b>Page500</b>	<b>Page500:</b> string; Set the 500 HTML page. The string will be sent as is.
<b>Page500File</b>	<b>Page500File:</b> string; Set the 500 HTML page file. The file content will be sent. This has higher precedence than <code>TBrookSettings.Page500</code> so when both are set, this will be processed first and only if the file is not found or cannot be read the system will fallback to <code>TBrookSettings.Page500</code>
<b>DirectoryForUploads</b>	<b>DirectoryForUploads:</b> string; Set the default directory for uploads.
<b>DeleteUploadedFiles</b>	<b>DeleteUploadedFiles:</b> Boolean; Defines if the temporary uploaded files will be deleted.
<b>KeepUploadedNames</b>	<b>KeepUploadedNames:</b> Boolean; Keeps the original name of the uploaded files.
<b>AcceptsJSONContent</b>	<b>AcceptsJSONContent:</b> Boolean; Defines if the application allows JSON formatted requests.
<b>Configuration</b>	<b>Configuration:</b> string; Set a configuration for the application or for its object members.

<b>RootUrl</b>	<code>RootUrl: string;</code> Set the default root URL. This is used by methods such as <code>TBrookAction.UrlFor</code> , <code>TBrookActionHelper.LinkTo</code> , <code>TBrookActionHelper.ButtonTo</code> etc. By default, Brook assumes <code>SCRIPT_NAME</code> as root URL.
<b>Port</b>	<code>Port: Word;</code> Set the default application port.
<b>OnError</b>	<code>OnError: TOnShowRequestException;</code> Handles the application exceptions.

## 16.5 Functions and Procedures

### BrookStartsChar

---

**Declaration** `function BrookStartsChar(const Ch: Char; const S: string): Boolean;`

**Description** Check whether a string starts with a given character.

### BrookEndsChar

---

**Declaration** `function BrookEndsChar(const Ch: Char; const S: string): Boolean;`

**Description** Check whether a string ends with a given character.

### BrookExtractPathLevels

---

**Declaration** `procedure BrookExtractPathLevels(S: string; var R: string; out ALvl: string; out AEndDelim: Boolean; const ADelimiter: Char = US);`

**Description** Get the next pathinfo level.

### BrookGetPathLevel

---

**Declaration** `function BrookGetPathLevel(const APath: string; const AIndex: SizeInt = 0; const ADelimiter: Char = US): string;`

**Description** Get the path level passing the respective index. Exemple:

`BrookGetPathLevel('/a/b/c/', 1) = b.`

### BrookGetPathLevels

---

**Declaration** `function BrookGetPathLevels(const APath: string; const AIndex: SizeInt = 0; const ADelimiter: Char = US): string;`

**Description** Get the path from the level correspondent to the index to the last level. Exemple:

`BrookGetPathLevels('/a/b/c/', 1) = b/c/.`

## BrookMatchMethod

---

**Declaration** `function BrookMatchMethod(const ABrookMethod: TBrookRequestMethod; const AMethod: string): Boolean;`

**Description** Checks if a string is equivalent an enumerator representing a HTTP request method.

## BrookFileDate

---

**Declaration** `function BrookFileDate(const AFileName: TFileName): TDateTime;`

**Description** Get the datetime of a file.

## BrookDumpStack

---

**Declaration** `function BrookDumpStack(const AEOL: ShortString = BR): string;`

**Description** Writes a backtrace of the current exception.

## BrookDumpStackTrace

---

**Declaration** `function BrookDumpStackTrace(const AEOL: ShortString = BR): string;`

**Description** Writes a stack trace of the current exception.

## BrookExcludeTrailingUrlDelimiter

---

**Declaration** `function BrookExcludeTrailingUrlDelimiter(const AUrl: string): string;`

**Description** Ensures Url ends without delimiter.

## BrookIncludeTrailingUrlDelimiter

---

**Declaration** `function BrookIncludeTrailingUrlDelimiter(const AUrl: string): string;`

**Description** Ensures Url ends with delimiter.

## BrookExists

---

**Declaration** `function BrookExists(const S: string; const AParts: array of string): Boolean; overload;`

**Description** Checks if a string exists in an array of strings.

## BrookExists

---

**Declaration** `function BrookExists(const S: string; const AParts: array of string; const AIgnoreCase: Boolean): Boolean; overload;`

**Description** Checks (ignoring case) if a string exists in an array of strings.



## BrookStringToObject

---

**Declaration** `procedure BrookStringToObject(AObject: TObject; APropInfo: PPropInfo; const AValue: string); overload;`

**Description** Fills a published property of an object passing the property as `PPropInfo` and value as `string`.

## BrookStringToObject

---

**Declaration** `procedure BrookStringToObject(AObject: TObject; const AName, AValue: string); overload;`

**Description** Fills a published property of an object passing the name and value as `string`.

## BrookSafeStringToObject

---

**Declaration** `procedure BrookSafeStringToObject(AObject: TObject; const AName, AValue: string);`

**Description** Fills a published property of an object passing the name and value as `string` and checking the params.

## BrookStringsToObject

---

**Declaration** `procedure BrookStringsToObject(AObject: TObject; AStrings: TStrings); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings.

## BrookStringsToObject

---

**Declaration** `procedure BrookStringsToObject(AObject: TObject; AStrings: TStrings; const AIgnoredProps: array of string); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings. Allows to ignore properties via an array of strings.

## BrookStringsToObject

---

**Declaration** `procedure BrookStringsToObject(AObject: TObject; AStrings: TStrings; const AIgnoredProps: TStrings); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings. Allows to ignore properties via a list of strings.

## BrookSafeStringsToObject

---

**Declaration** `procedure BrookSafeStringsToObject(AObject: TObject; AStrings: TStrings); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings and checking the params.

## BrookSafeStringsToObject

---

**Declaration** `procedure BrookSafeStringsToObject(AObject: TObject; AStrings: TStrings; const AIgnoredProps: array of string); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings and checking the params. Allows to ignore properties via an array of strings.

## BrookSafeStringsToObject

---

**Declaration** `procedure BrookSafeStringsToObject(AObject: TObject; AStrings: TStrings; const AIgnoredProps: TStrings); overload;`

**Description** Fills the published properties of an object passing the names and values as a list of strings and checking the params. Allows to ignore properties via a list of strings.

## BrookObjectToString

---

**Declaration** `procedure BrookObjectToString(AObject: TObject; APropInfo: PPropInfo; out AValue: string); overload;`

**Description** Reads a published property of an object passing the property as PPropInfo and getting the value as string.

## BrookObjectToString

---

**Declaration** `procedure BrookObjectToString(AObject: TObject; const AName: string; out AValue: string); overload;`

**Description** Reads a published property of an object passing the name as string and getting the value as string.

## BrookSafeObjectToString

---

**Declaration** `procedure BrookSafeObjectToString(AObject: TObject; const AName: string; out AValue: string);`

**Description** Reads a published property of an object passing the name, getting the value as string and checking the params.

## BrookObjectToStrings

---

**Declaration** `procedure BrookObjectToStrings(AObject: TObject; AStrings: TStrings); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings.

## BrookObjectToStrings

---

**Declaration** `procedure BrookObjectToStrings(AObject: TObject; AStrings: TStrings; const AIgnoredProps: array of string); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings. Allows to ignore properties via an array of strings.

## BrookObjectToStrings

---

**Declaration** `procedure BrookObjectToStrings(AObject: TObject; AStrings: TStrings; const AIgnoredProps: TStrings); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings. Allows to ignore properties via a list of strings.

## BrookSafeObjectToStrings

---

**Declaration** `procedure BrookSafeObjectToStrings(AObject: TObject; AStrings: TStrings); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings and checking the params.

## BrookSafeObjectToStrings

---

**Declaration** `procedure BrookSafeObjectToStrings(AObject: TObject; AStrings: TStrings; const AIgnoredProps: array of string); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings and checking the params. Allows to ignore properties via an array of strings.

## BrookSafeObjectToStrings

---

**Declaration** `procedure BrookSafeObjectToStrings(AObject: TObject; AStrings: TStrings; const AIgnoredProps: TStrings); overload;`

**Description** Reads the published properties of an object getting the names and values as a list of strings and checking the params.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject; const AIgnoredProps: array of string); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count. Allows to ignore properties via an array of strings.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject; const AIgnoredProps: TStrings); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count. Allows to ignore properties via a list of strings.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(AFrom, ATo: TObject); overload;`

**Description** Copies the value of all properties from one object to another.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(AFrom, ATo: TObject; const AIgnoredProps: array of string); overload;`

**Description** Copies the value of all properties from one object to another. Allows to ignore properties via an array of strings.

## BrookCopyObject

---

**Declaration** `procedure BrookCopyObject(AFrom, ATo: TObject; const AIgnoredProps: TStrings); overload;`

**Description** Copies the value of all properties from one object to another. Allows to ignore properties via a list of strings.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count and checking the params.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject; const AIgnoredProps: array of string); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count and checking the params. Allows to ignore properties via an array of strings.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(APropList: PPropList; const APropCount: Integer; AFrom, ATo: TObject; const AIgnoredProps: TStrings); overload;`

**Description** Copies the value of all properties from one object to another passing the prop. list and prop. count and checking the params. Allows to ignore properties via a list of strings.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(AFrom, ATo: TObject); overload;`

**Description** Copies the value of all properties from one object to another and checking the params.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(AFrom, ATo: TObject; const AIgnoredProps: array of string); overload;`

**Description** Copies the value of all properties from one object to another and checking the params. Allows to ignore properties via an array of strings.

## BrookSafeCopyObject

---

**Declaration** `procedure BrookSafeCopyObject(AFrom, ATo: TObject; const AIgnoredProps: TStrings); overload;`

**Description** Copies the value of all properties from one object to another and checking the params. Allows to ignore properties via a list of strings.

## 16.6 Types

### TBrookArrayOfString

---

**Declaration** `TBrookArrayOfString = array of string;`

**Description** Defines an array of strings.

## TBrookRequestMethod

---

**Declaration** TBrookRequestMethod = (...);

**Description** Defines an enumerator to represent the HTTP request methods.

**Values** rmUnknown  
rmAll  
rmGet  
rmPost  
rmPut  
rmDelete  
rmHead  
rmOptions  
rmTrace

## 16.7 Variables

### BrookSettings

---

**Declaration** BrookSettings: TBrookSettings = ( Mapped: False; Charset: BROOK\_HTTP\_CHARSET\_UTF\_8; ContentType: BROOK\_HTTP\_CONTENT\_TYPE\_TEXT\_HTML; Page404: BROOK\_HTTP\_RESPONSE\_TEMPLATE\_NOT\_FOUND; Page404File: ES; Page500: BROOK\_HTTP\_RESPONSE\_TEMPLATE\_INTERNAL\_SERVER\_ERROR; Page500File: ES; DirectoryForUploads: ES; DeleteUploadedFiles: False; KeepUploadedNames: True; AcceptsJSONContent: False; Configuration: ES; RootUrl: ES; Port: 0; OnError: nil; );

**Description** Global variable to store Brook settings.