Brook for Free Pascal

Silvio Clecio, Luciano Souza

January 24, 2019

Contents

Chapter 1

Unit BrookAction

1.1 Description

Action classes.

1.2 Uses

- `BrookClasses`(3)
- `BrookHttpDefs`(10)
- `BrookException`(7)
- `BrookMessages`(13)
- `BrookUtils`(17)
- `BrookConsts`(6)
- `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 > `TBrookPersistent`(3.4) > TPersistent

Description

Provides features to handle HTTP requests and responses.

Properties

Field `public property Field[constAName:string]: string read GetField write SetField;`

Handles the `fields`(1.4) of a form.

Param `public property Param[constAName:string]: string read GetParam write SetParam;`

Handles the Query_String parameters of a URL.

Variable `public property Variable[constAName:string]: string read GetVariable write SetVariable;`

Handles `variables`(1.4) from a parametrized URL.

Files `public property Files: TBrookUploadedFiles read FFiles;`

Handles a file list of `fields`(1.4) of a form.

Fields `public property Fields: TStrings read FFields;`

Handles a string list of `fields` of a form.

Params `public property Params: TStrings read FParams;`

Handles a string list of the Query_String parameters of a URL.

Variables `public property Variables: TStrings read FVariables;`

Handles a string list of `variables` from a parametrized URL.

Method `public property Method: string read GetMethod;`

Returns the HTTP `request`(1.4) `method`.

HttpRequest `public property HttpRequest: TBrookRequest read FHttpRequest;`

Provides services related to HTTP requests drived to a webserver.

HttpResponse `public property HttpResponse: TBrookResponse read FHttpResponse;`

Provides services related to the HTTP responses comming back from a webserver.

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`(1.4) to `request`(1.4)/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`(1.4) inside URLs:

:name – Represents a `variable`(1.4) that spans single URL component between slashes.
Examples:
`TMyAction.Register('/foo/:myvar');`
Value of a `variable`(1.4) `"myvar"` can be read from the property `Variables`(1.4) or `Variable`(1.4), e.g.:
`Write(1.4)(Variables.Values['myvar']);`
`Write(1.4)(Variable(1.4)['myvar']);`
Any number of `variables`(1.4) can be combined:
`TMyAction.Register('/foo/:cat/:id');`

*name – **Represents a `variable`(1.4) that spans one or more levels between slashes in the c**

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`(1.4) `Variables.Values['path']` will receive `'file'`, `'dir/file'` or `'dir/subdir/file'` correspondingly.

You can also add static text after `variable`(1.4) 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`(1.4) can be combined with any number of single-level `variables`(1.4) in any order:

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

NOTE: Only one multi-level `variable`(1.4) 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`(1.4) 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`(1.4) `method`(1.4).

Parameters **AMethod** Informs the HTTP `request`(1.4) `method`(1.4) being valid the following `options`(1.4): `rmAll(17.6)`, `rmGet(17.6)`, `rmHead(17.6)`, `rmOptions(17.6)`, `rmPost(17.6)`, `rmPut(17.6)` or `rmDelete(17.6)`. The only way to `register` two actions with the same pattern is differentiating the value of this parameter. If `at`(6.2) least one action has this parameter changed, the route mapping is enabled in `TBrookSettings`(17.4).`Mapped`(17.4). A typical example of use is:

```
@(longcode
 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`.

DoFillVariables

Declaration `public procedure DoFillVariables(const ANames, AValues: TBrookArrayOfString); virtual;`

Description Fills the `Variables`(1.4) with the registered `variables`(1.4) passed through the URL.

DoFillingVariables

Declaration `public procedure DoFillingVariables(const AIndex: Integer; const ANames, AValues: TBrookArrayOfString); virtual;`

Description Fills the `Variables`(1.4) with the registered `variables`(1.4) passed one by one through the URL.

SetCookie

Declaration `public procedure SetCookie(const AName, AValue:  string; const AExpires:  TDateTime = NullDate; const APath:  string = ES; const ADomain:  string = ES; const ASecure:  Boolean = False; const AHttpOnly:  Boolean = False);`

Description Creates a cookie.

GetCookie

Declaration `public function GetCookie(const AName:  string):  string;`

Description `Get`(1.4) a cookie value.

DeleteCookie

Declaration `public procedure DeleteCookie(const AName:  string; const APath:  string = ES; const ADomain:  string = ES);`

Description Deletes a cookie.

DoRequest

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

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

Request

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

Description Is triggered by a `request` of any HTTP `method`(1.4).

GetFields

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

Description `Get`(1.4) an object with the `fields`(1.4) coming from a `x-www-form-urlencoded` form.

GetParams

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

Description `Get`(1.4) an object with the `params`(1.4) coming from a `QUERY_STRING`.

GetVariables

Declaration `public procedure GetVariables(AObject: TObject);`

Description `Get`(1.4) an object with the `variables`(1.4) 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:

```
@(longcode
       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`(1.4) `method`(1.4).

Post

Declaration `public procedure Post; virtual;`

Description Is triggered by a `POST` HTTP `request`(1.4) `method`(1.4).

Put

Declaration `public procedure Put; virtual;`

Description Is triggered by a `PUT` HTTP `request`(1.4) `method`(1.4).

Patch

Declaration `public procedure Patch; virtual;`

Description Is triggered by a `PATCH` HTTP `request`(1.4) `method`(1.4).

Delete

Declaration `public procedure Delete; virtual;`

Description Is triggered by a `DELETE` HTTP `request`(1.4) `method`(1.4).

Head

Declaration `public procedure Head; virtual;`

Description Is triggered by a `HEAD` HTTP `request`(1.4) `method`(1.4).

Options

Declaration `public procedure Options; virtual;`

Description Is triggered by an `OPTIONS` HTTP `request`(1.4) `method`(1.4).

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 formated 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(1.4)()`, `WriteLn()`, `Render(1.4)()` etc.

Exists

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

Description Checks if a name `exists` in `fields`(1.4).

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(AObject: TObject; const AIgnoredProps: TStrings); overload;`

Description Writes an object allowing to ignore properties via an array of strings.

Write

Declaration `public procedure Write(AObject: TObject; const AIgnoredProps: array of string); overload;`

Description Writes an object allowing to ignore properties via a list of strings.

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) > `TBrookPersistent`(3.4) > TPersistent

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`(1.4).

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

`Application`(2.5) interface.

2.2 Uses

- `BrookConsts`(6)
- `BrookClasses`(3)
- `BrookException`(7)
- `BrookMessages`(13)
- `Classes`

2.3 Overview

`EBrookApplication Class` Handles exceptions for `application`(2.5).

`IBrookApplication Interface` Interface of the `application`(2.5).

`BrookApp` Returns the `application`(2.5) instance.

`Application` Returns the `application` instance maintaining compatibility with legacy code.

`BrookRegisterApp` Register the `application`(2.5).

`BrookUnregisterApp` Unregister the `application`(2.5).

2.4 Classes, Interfaces, Objects and Records

EBrookApplication Class

Hierarchy

EBrookApplication > `EBrook`(7.4) > Exception

Description

Handles exceptions for `application`(2.5).

IBrookApplication Interface

Hierarchy

IBrookApplication > `IBrookInterface`(3.4) > IInterface

Description

Interface of the `application`(2.5).

Properties

Terminated `public property Terminated: Boolean read GetTerminated;`

Checks whether the `application`(2.5) is `terminated`.

Methods

CreateForm

Declaration `public procedure CreateForm(AInstanceClass: TComponentClass; out AReference);`

Description Creates an `instance`(2.4) of the type specified by the `AInstanceClass` parameter and assigns it to the variable given by the `AReference` parameter. The owner of the new `instance`(2.4) is the `Application(2.5)` object.

GetTerminated

Declaration `public function GetTerminated: Boolean;`

Description Gets whether the `application`(2.5) is `terminated`(2.4).

Instance

Declaration `public function Instance: TObject;`

Description Returns the `instance` of broker `application`(2.5).

Run

Declaration `public procedure Run;`

Description Initializes and runs the `application`(2.5).

Terminate

Declaration `public procedure Terminate;`

Description Terminates the `application`(2.5).

2.5 Functions and Procedures

BrookApp

Declaration `function BrookApp: IBrookApplication;`

Description Returns the `application`(2.5) instance.

Application

Declaration `function Application: IBrookApplication;`

Description Returns the `application` instance maintaining compatibility with legacy code.

BrookRegisterApp

Declaration `procedure BrookRegisterApp(AApp: IBrookApplication);`

Description Register the `application`(2.5).

BrookUnregisterApp

Declaration `procedure BrookUnregisterApp;`

Description Unregister the `application`(2.5).

Chapter 3

Unit BrookClasses

3.1 Description

Base classes.

3.2 Uses

- `BrookConsts`(6)
- `Classes`
- `RtlConsts`

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.

`TBrookPersistent Class` Is the main persistent for Brook.

`TBrookComponent Class` Is the main component for Brook.

`TBrookDataModule Class` Is the main data module 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.

TBrookPersistent Class

Hierarchy

TBrookPersistent > TPersistent

Description

Is the main persistent for Brook.

TBrookComponent Class

Hierarchy

TBrookComponent > TComponent

Description

Is the main component for Brook.

TBrookDataModule Class

Hierarchy

TBrookDataModule > TDataModule

Description

Is the main data module for Brook.

Methods

Create

Declaration `public constructor Create(AOwner:  TComponent); override;`

3.5 Types

TBrookClass

Declaration `TBrookClass = class of TBrookObject;`

Description Is the main class for Brook.

Chapter 4

Unit BrookConfigurator

4.1 Description

Configurator class.

4.2 Uses

- `BrookClasses`(3)
- `BrookUtils`(17)
- `BrookException`(7)
- `BrookMessages`(13)
- `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 > `TBrookComponent`(3.4) > TComponent

Description

Configures objects by means of string or file.

Properties

Target `public property Target: TObject read GetTarget write SetTarget;`

Defines the object to be configured.

Prop `public property Prop[constAName:string]: string read GetProp write SetProp;`

Handles the `target`(4.4) properties.

Param `public property Param[constAName:string]: string read GetParam write SetParam;`

Handles a string list of `params`(4.4) of a configuration.

IgnoredParams `public property IgnoredParams: TStrings read FIgnoredParams write SetIgnoredParams;`

Ignored `params`(4.4) in the configuration.

Params `public property Params: TStrings read FParams write SetParams;`

`Params` of the configuration.

AfterConfigure `public property AfterConfigure: TBrookConfigureEvent read FAfterConfigure write FAfterConfigure;`

Is triggered after `configure`(4.4).

BeforeConfgure `public property BeforeConfgure: TBrookConfigureEvent read FBeforeConfgure write FBeforeConfgure;`

Is triggered before `configure`(4.4).

Methods

Create

Declaration `public constructor Create(AOwner: TComponent); override;`

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

Destroy

Declaration `public destructor Destroy; override;`

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

Configure

Declaration `public procedure Configure;`

Description Configures the `target`(4.4) property.

4.5 Types

TBrookConfiguratorClass

Declaration `TBrookConfiguratorClass = class of TBrookConfigurator;`

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

TBrookConfigureEvent

Declaration `TBrookConfigureEvent = procedure(ASender:  TObject; var AHandled:  Boolean) of object;`

Description Is a type to configure event.

PBrookConfigureEvent

Declaration `PBrookConfigureEvent = ^TBrookConfigureEvent;`

Description Defines a pointer to the configure event.

Chapter 5

Unit BrookConstraints

5.1 Description

Constraints unit.

5.2 Uses

- `BrookClasses`(3)
- `BrookAction`(1)
- `BrookRouter`(15)
- `BrookException`(7)
- `BrookMessages`(13)
- `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(5.4)` with a formated 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 formated 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 `public ActionClass: TBrookActionClass;`

ConstraintClass `public 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`(5.4) provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the `service`(5.4) 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

Consts unit.

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-95C3-664CDFFDAA9E}’;`

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%.2x’;`

BROOK_UUID_SEP_MASK

Declaration `BROOK_UUID_SEP_MASK = ’%.8x-%.4x-%.4x-%.2x%.2x-%.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’, ’.cdbcmsg’), (’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) - Contaimer’, ’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-Seperated 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.uvvu.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 Siesmograph 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
Languate’, ’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.medcalcdata’, ’.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’, ’.mfm’), (’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’), (’Micosoft PowerPoint - Macro-Enabled Template File’,
’application/vnd.ms-powerpoint.template.macroenabled.12’, ’.potm’),
(’Micosoft Word - Macro-Enabled Document’,
’application/vnd.ms-word.document.macroenabled.12’, ’.docm’), (’Micosoft 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', '.sxg'), ('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
(Formely PCL XL)', 'application/vnd.hp-pclxl', '.pclxl'), ('Pcsel eFIF File',
'application/vnd.picsel', '.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.semf’, ’.semf’), (’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 Seperated 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-uuencode', '.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/winhlp’, ’.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’), (’XPInstall - 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;`

DefCurrency

Declaration `DefCurrency: Currency = 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

Exceptions classes.

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(7.4)`.

Create

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

Description Creates an instance of `EBrook(7.4)`.

Create

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

Description Creates an instance of `EBrook(7.4)`.

CreateFmt

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

Description Creates an instance of `EBrook(7.4)` with a formated message.

CreateFmt

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

Description Creates an instance of `EBrook(7.4)` with a formated message.

CreateFmt

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

Description Creates an instance of `EBrook(7.4)` with a formated 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

HTTP client classes.

8.2 Uses

- `BrookClasses`(3)
- `BrookException`(7)
- `BrookMessages`(13)
- `BrookConsts`(6)
- `BrookHttpConsts`(9)
- `Classes`
- `SysUtils`
- `FGL`

8.3 Overview

`EBrookHttpDef Class` Handles exceptions for `TBrookHttpDef`(8.4).

`EBrookHttpDefs Class` Handles exceptions for `TBrookHttpDefs`(8.4).

`EBrookHttpClient Class` Handles exceptions for `EBrookHttpClient`(8.4).

`TBrookHttpResult 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).

TBrookHttpResult Record

Description

Information returned after a request.

Fields

StatusCode	`public StatusCode: Integer;`
ReasonPhrase	`public ReasonPhrase: string;`
Header	`public Header: string;`
Content	`public 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`(8.4).

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`(8.4).

Headers `public property Headers: TStrings read GetHeaders;`

`Request`(8.4) `headers`.

StatusCode `public property StatusCode: Integer read GetStatusCode;`

Result code after successful `request`(8.4).

ReasonPhrase `public property ReasonPhrase: string read GetReasonPhrase;`

Result text after successful `request`(8.4).

Method `public property Method: string read GetMethod write SetMethod;`

`Method` for requests.

Url `public property Url: string read GetUrl write SetUrl;`

`URL` that `request`(8.4) 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`(8.4) 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`(8.4) by a `GET` HTTP `request`(8.4) `method`(8.4).

Post

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

Description Sends `request`(8.4) by a `POST` HTTP `request`(8.4) `method`(8.4).

Put

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

Description Sends `request`(8.4) by a `PUT` HTTP `request`(8.4) `method`(8.4).

Delete

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

Description Sends `request`(8.4) by a `DELETE` HTTP `request`(8.4) `method`(8.4).

Options

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

Description Sends `request`(8.4) by an `OPTIONS` HTTP `request`(8.4) `method`(8.4).

Head

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

Description Sends `request`(8.4) by a `HEAD` HTTP `request`(8.4) `method`(8.4).

PostForm

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) `method`(8.4), passing a form-data as parameter.

PostForm

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) `method`(8.4), passing a form-data as parameter.

PutForm

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

Description Sends `request`(8.4) by a `PUT`(8.4) HTTP `request`(8.4) `method`(8.4), passing a form-data as parameter.

PutForm

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

Description Sends `request`(8.4) by a `PUT`(8.4) HTTP `request`(8.4) `method`(8.4), 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`(8.4) by a `POST`(8.4) HTTP `request`(8.4) `method`(8.4), passing a file as parameter.

PostFile

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) `method`(8.4), 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(8.4)`.

Get

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

Description Sends `request`(8.4) by a `GET` HTTP `request`(8.4) method.

Post

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

Description Sends `request`(8.4) by a `POST` HTTP `request`(8.4) method.

Put

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

Description Sends `request`(8.4) by a `PUT` HTTP `request`(8.4) method.

Delete

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

Description Sends `request`(8.4) by a `DELETE` HTTP `request`(8.4) method.

Options

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

Description Sends `request`(8.4) by an `OPTIONS` HTTP `request`(8.4) method.

Head

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

Description Sends `request`(8.4) by a `HEAD` HTTP `request`(8.4) method.

PostForm

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) method, passing a form-data as parameter.

PostForm

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) method, passing a form-data as parameter.

PutForm

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

Description Sends `request`(8.4) by a `PUT`(8.4) HTTP `request`(8.4) method, passing a form-data as parameter.

PutForm

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

Description Sends `request`(8.4) by a `PUT`(8.4) HTTP `request`(8.4) 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`(8.4) by a `POST`(8.4) HTTP `request`(8.4) method, passing a file as parameter.

PostFile

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

Description Sends `request`(8.4) by a `POST`(8.4) HTTP `request`(8.4) method, passing a file as parameter.

Request

Declaration `public function Request(AHttp: TBrookHttpDef): TBrookHttpResult;`

Description Performs the `request`(8.4), passing a `TBrookHttpDef(8.4)` as parameter.

Request

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

Description Performs the `request`(8.4), passing the method and URL as parameter.

Request

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

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

HTTP consts unit.

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</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

HttpDefs classes.

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`.

TBrookCookie

Declaration `TBrookCookie = TCookie;`

Description Alias to `TCookie`.

TBrookCookies

Declaration `TBrookCookies = TCookies;`

Description Alias to `TCookies`.

Chapter 11

Unit BrookHttpUtils

11.1 Description

HTTP utilities unit.

11.2 Uses

- `BrookHttpClient`(8)
- `BrookMessages`(13)
- `BrookHttpConsts`(9)
- `BrookConsts`(6)
- `BrookUtils`(17)
- `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 AceptEnconding header.

`BrookGetAcceptEncoding` Returns a string of HTTP AcceptEnconding.

`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(17.6)`.

`BrookStrToRequestMethod` Returns the `TBrookRequestMethod(17.6)` 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 AceptEnconding header.

BrookGetAcceptEncoding

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

Description Returns a string of HTTP AcceptEnconding.

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(17.6)`.

BrookStrToRequestMethod

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

Description Returns the `TBrookRequestMethod(17.6)` 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 BrookLogger

12.1 Description

Logger classes.

12.2 Uses

- `BrookClasses`(3)
- `BrookException`(7)
- `BrookMessages`(13)
- `SysUtils`
- `Classes`

12.3 Overview

`EBrookLogger Class` Handles exceptions for `TBrookLogger`(12.4).

`TBrookLogger Class` Provides features for the `application`(2.5) logging.

12.4 Classes, Interfaces, Objects and Records

EBrookLogger Class

Hierarchy

EBrookLogger > `EBrook`(7.4) > Exception

Description

Handles exceptions for `TBrookLogger`(12.4).

TBrookLogger Class

Hierarchy

TBrookLogger > `TBrookComponent`(3.4) > TComponent

Description

Provides features for the `application`(2.5) logging.

Properties

Active `public property Active: Boolean read GetActive write SetActive;`

Enables or disables the logger.

FileName `public property FileName: TFileName read GetFileName write SetFileName;`

Defines the name of the `log`(12.4) file.

Types `public property Types: TBrookLogTypes read FTypes write FTypes;`

The logger `output`(12.4) `types`.

Output `public property Output: TBrookLogOutput read GetOutput write SetOutput;`

The logger `output` mode.

Prepared `public property Prepared: Boolean read FPrepared;`

Return `True` if broker is `prepared`.

AfterLog `public property AfterLog: TBrookLogEvent read FAfterLog write FAfterLog;`

Is triggered after the logger writes a `log`(12.4).

BeforeLog `public property BeforeLog: TBrookLogEvent read FBeforeLog write FBeforeLog;`

Is triggered before the logger writes a `log`(12.4).

Methods

Create

Declaration `public constructor Create(AOwner: TComponent); override;`

GetServiceClass

Declaration `public class function GetServiceClass: TBrookLoggerClass;`

Description Return the `service`(12.4) class provided by this class.

RegisterService

Declaration `public class procedure RegisterService;`

Description Registers the `service`(12.4) provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the `service`(12.4) provided by this class.

Service

Declaration `public class function Service: TBrookLogger;`

Description Return an instance of this class.

Prepare

Declaration `public procedure Prepare; virtual;`

Description `Prepare` the logger broker.

Unprepare

Declaration `public procedure Unprepare; virtual;`

Description `Unprepare` the logger broker.

Log

Declaration `public procedure Log(const AType: TBrookLogType; const S: string; const ACode: Word; const E: Exception = nil); virtual; abstract;`

Description Writes a `log`.

DoLog

Declaration `public procedure DoLog(const AType: TBrookLogType; const S: string; const ACode: Word; const E: Exception = nil); virtual;`

Description Writes a `log`(12.4) triggering the `AfterLog(12.4)` and (`BeforeLog`(12.4)) events.

Custom

Declaration `public procedure Custom(const S: string; const ACode: Word); virtual;`

Description Writes a `custom` `log`(12.4).

Info

Declaration `public procedure Info(const S: string); virtual;`

Description Writes an information `log`(12.4).

Warn

Declaration `public procedure Warn(const S: string); virtual;`

Description Writes a warning `log`(12.4).

Debug

Declaration `public procedure Debug(const S: string); virtual;`

Description Writes a `debug` `log`(12.4).

Error

Declaration `public procedure Error(const S: string; E: Exception = nil); virtual;`

Description Writes an `error` `log`(12.4).

12.5 Types

TBrookLoggerClass

Declaration `TBrookLoggerClass = class of TBrookLogger;`

Description Is a metaclass for `TBrookLogger`(12.4) class.

TBrookLogOutput

Declaration `TBrookLogOutput = (...);`

Description Defines an enumerator to represent the logger output kind.

Values `loFile`
`loSystem`

TBrookLogType

Declaration `TBrookLogType = (...);`

Description Defines an enumerator to represent the logger event types.

Values `ltCustom`
`ltInfo`
`ltWarning`
`ltError`
`ltDebug`

TBrookLogTypes

Declaration `TBrookLogTypes = set of TBrookLogType;`

Description Defines a set to represent the logger event types.

TBrookLogEvent

Declaration

```
TBrookLogEvent = procedure(ASender:  TObject; const AType:  TBrookLogType;
const S: string; const ACode:  Word; const E: Exception; var AHandled:
Boolean) of object;
```

Description Is a type to the log event.

PBrookLogEvent

Declaration `PBrookLogEvent = ^TBrookLogEvent;`

Description Defines a pointer to the log event.

Chapter 13

Unit BrookMessages

13.1 Description

Messages unit.

13.2 Constants

SBrookDefaultLocale_rst

Declaration `SBrookDefaultLocale_rst = 'en-US';`

SBrookInvalidRequestMethodError_rst

Declaration `SBrookInvalidRequestMethodError_rst = 'Invalid request method:  %s';`

SBrookItemNotFoundError_rst

Declaration `SBrookItemNotFoundError_rst = 'Item "%s" not found.';`

SBrookFileNotFoundError_rst

Declaration `SBrookFileNotFoundError_rst = 'File not found:  %s';`

SBrookNoRequestMethodError_rst

Declaration `SBrookNoRequestMethodError_rst = 'No REQUEST_METHOD passed from server.';`

SBrookNoApplicationRegisteredError_rst

Declaration `SBrookNoApplicationRegisteredError_rst = 'No application registered.';`

SBrookApplicationAlreadyRegisteredError_rst

Declaration `SBrookApplicationAlreadyRegisteredError_rst = ’The application is already registered.’;`

SBrookRegiterTBrookActionError_rst

Declaration `SBrookRegiterTBrookActionError_rst = ’Unable to register the type TBrookAction directly.’;`

SBrookActionAlreadyRegisteredError_rst

Declaration `SBrookActionAlreadyRegisteredError_rst = ’The action "%s" is already registered.’;`

SBrookDefaultActionAlreadyRegisteredError_rst

Declaration `SBrookDefaultActionAlreadyRegisteredError_rst = ’There is already a default action registered.’;`

SBrookPatternAlreadyRegisteredError_rst

Declaration `SBrookPatternAlreadyRegisteredError_rst = ’There is already an action registered with the pattern "%s".’;`

SBrookRouterServiceAlreadyRegisteredError_rst

Declaration `SBrookRouterServiceAlreadyRegisteredError_rst = ’The router service is already registered.’;`

SBrookNoRouterServiceRegisteredError_rst

Declaration `SBrookNoRouterServiceRegisteredError_rst = ’No router service registered.’;`

SBrookNoRouteRegisteredError_rst

Declaration `SBrookNoRouteRegisteredError_rst = ’No route registered.’;`

SBrookCfgFileNotFoundError_rst

Declaration `SBrookCfgFileNotFoundError_rst = ’The config file was not found:  "%s"’;`

SBrookNotNilError_rst

Declaration `SBrookNotNilError_rst = ’"%s" must not be nil.’;`

SBrookEmptyLibraryNameError_rst

Declaration `SBrookEmptyLibraryNameError_rst = ’The library name must not be empty.’;`

SBrookMethodNotAllowedError_rst

Declaration SBrookMethodNotAllowedError_rst = ’HTTP method not allowed for the requested resource.’;

SBrookConstraintAlreadyRegisteredError_rst

Declaration SBrookConstraintAlreadyRegisteredError_rst = ’The constraint "%s" is already registered.’;

SBrookConstraintsServiceAlreadyRegisteredError_rst

Declaration SBrookConstraintsServiceAlreadyRegisteredError_rst = ’The constraints service is already registered.’;

SBrookNoConstraintsServiceRegisteredError_rst

Declaration SBrookNoConstraintsServiceRegisteredError_rst = ’No constraints service registered.’;

SBrookNoLoggerServiceRegisteredError_rst

Declaration SBrookNoLoggerServiceRegisteredError_rst = ’No logger service registered.’;

SBrookLoggerServiceAlreadyRegisteredError_rst

Declaration SBrookLoggerServiceAlreadyRegisteredError_rst = ’The logger service is already registered.’;

13.3 Variables

SBrookDefaultLocale

Declaration SBrookDefaultLocale: string = SBrookDefaultLocale_rst;

SBrookInvalidRequestMethodError

Declaration SBrookInvalidRequestMethodError: string = SBrookInvalidRequestMethodError_rst;

SBrookItemNotFoundError

Declaration SBrookItemNotFoundError: string = SBrookItemNotFoundError_rst;

SBrookFileNotFoundError

Declaration SBrookFileNotFoundError: string = SBrookFileNotFoundError_rst;

SBrookNoRequestMethodError

Declaration

```
SBrookNoRequestMethodError: string = SBrookNoRequestMethodError_rst;
```

SBrookNoApplicationRegisteredError

Declaration

```
SBrookNoApplicationRegisteredError: string =
SBrookNoApplicationRegisteredError_rst;
```

SBrookApplicationAlreadyRegisteredError

Declaration

```
SBrookApplicationAlreadyRegisteredError: string =
SBrookApplicationAlreadyRegisteredError_rst;
```

SBrookRegiterTBrookActionError

Declaration

```
SBrookRegiterTBrookActionError: string = SBrookRegiterTBrookActionError_rst;
```

SBrookActionAlreadyRegisteredError

Declaration

```
SBrookActionAlreadyRegisteredError: string =
SBrookActionAlreadyRegisteredError_rst;
```

SBrookDefaultActionAlreadyRegisteredError

Declaration

```
SBrookDefaultActionAlreadyRegisteredError: string =
SBrookDefaultActionAlreadyRegisteredError_rst;
```

SBrookPatternAlreadyRegisteredError

Declaration

```
SBrookPatternAlreadyRegisteredError: string =
SBrookPatternAlreadyRegisteredError_rst;
```

SBrookRouterServiceAlreadyRegisteredError

Declaration

```
SBrookRouterServiceAlreadyRegisteredError: string =
SBrookRouterServiceAlreadyRegisteredError_rst;
```

SBrookNoRouterServiceRegisteredError

Declaration

```
SBrookNoRouterServiceRegisteredError: string =
SBrookNoRouterServiceRegisteredError_rst;
```

SBrookNoRouteRegisteredError

Declaration

```
SBrookNoRouteRegisteredError: string = SBrookNoRouteRegisteredError_rst;
```

SBrookCfgFileNotFoundError

Declaration `SBrookCfgFileNotFoundError: string = SBrookCfgFileNotFoundError_rst;`

SBrookNotNilError

Declaration `SBrookNotNilError: string = SBrookNotNilError_rst;`

SBrookEmptyLibraryNameError

Declaration `SBrookEmptyLibraryNameError: string = SBrookEmptyLibraryNameError_rst;`

SBrookMethodNotAllowedError

Declaration `SBrookMethodNotAllowedError: string = SBrookMethodNotAllowedError_rst;`

SBrookConstraintAlreadyRegisteredError

Declaration

```
SBrookConstraintAlreadyRegisteredError: string =
SBrookConstraintAlreadyRegisteredError_rst;
```

SBrookConstraintsServiceAlreadyRegisteredError

Declaration

```
SBrookConstraintsServiceAlreadyRegisteredError: string =
SBrookConstraintsServiceAlreadyRegisteredError_rst;
```

SBrookNoConstraintsServiceRegisteredError

Declaration

```
SBrookNoConstraintsServiceRegisteredError: string =
SBrookNoConstraintsServiceRegisteredError_rst;
```

SBrookNoLoggerServiceRegisteredError

Declaration

```
SBrookNoLoggerServiceRegisteredError: string =
SBrookNoLoggerServiceRegisteredError_rst;
```

SBrookLoggerServiceAlreadyRegisteredError

Declaration

```
SBrookLoggerServiceAlreadyRegisteredError: string =
SBrookLoggerServiceAlreadyRegisteredError_rst;
```

Chapter 14

Unit BrookMiddleware

14.1 Description

Middleware classes.

14.2 Uses

- `BrookClasses`(3)
- `BrookAction`(1)
- `BrookRouter`(15)
- `BrookException`(7)
- `BrookHttpDefs`(10)
- `BrookUtils`(17)

14.3 Overview

`EBrookMiddleware Class` Handles exceptions for `TBrookMiddleware`(14.4).

`TBrookMiddleware Class` Intermediates two classes through a `TBrookExecuteActionEvent(15.5)` event.

14.4 Classes, Interfaces, Objects and Records

EBrookMiddleware Class

Hierarchy

EBrookMiddleware > `EBrook`(7.4) > Exception

Description

Handles exceptions for `TBrookMiddleware`(14.4).

TBrookMiddleware Class

Hierarchy

TBrookMiddleware > `TBrookComponent`(3.4) > TComponent

Description

Intermediates two classes through a `TBrookExecuteActionEvent(15.5)` event.

Properties

OnExecute `public property OnExecute: TBrookMiddlewareExecuteEvent read FOnExecute write FOnExecute;`

Is triggered when the `Execute(14.4)` method bound in this class is executed.

Methods

Create

Declaration `public constructor Create(ABoundEvent: PBrookExecuteActionEvent); overload; virtual;`

Description Creates an instance of a `TBrookMiddleware`(14.4) class.

Execute

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

Description Is triggered when the `DoExecute` method bound in this class is executed.

BindExecution

Declaration `public procedure BindExecution(AEvent: PBrookExecuteActionEvent);`

Description Bindes a `TBrookExecuteActionEvent(15.5)` event to this class keeping the implementation of a previously declared event.

14.5 Types

TBrookMiddlewareClass

Declaration `TBrookMiddlewareClass = class of TBrookMiddleware;`

Description Is a metaclass for `TBrookMiddleware`(14.4) class.

TBrookMiddlewareExecuteEvent

Declaration `TBrookMiddlewareExecuteEvent = procedure(ASender: TObject; AAction: TBrookAction; ARoute: TBrookRoute) of object;`

Description Is a type to `*Middleware.OnExecute` event.

PBrookMiddlewareExecuteEvent

Declaration `PBrookMiddlewareExecuteEvent = ^TBrookMiddlewareExecuteEvent;`

Description Defines a pointer to the `TBrookMiddlewareExecuteEvent(14.5)` type.

Chapter 15

Unit BrookRouter

15.1 Description

Router classes.

15.2 Uses

- `BrookClasses`(3)
- `BrookHttpDefs`(10)
- `BrookException`(7)
- `BrookAction`(1)
- `BrookUtils`(17)
- `BrookConsts`(6)
- `BrookMessages`(13)
- `BrookHTTPConsts`(9)
- `HTTPDefs`
- `Classes`
- `SysUtils`
- `StrUtils`

15.3 Overview

`EBrookRoutes Class` Handles exceptions for `TBrookRoutes`(15.4).

`EBrookRouter Class` Handles exceptions for `TBrookRouter`(15.4).

`TBrookRoute Record` Defines a route item.

`TBrookRoutes Class` Defines a list of routes.

`TBrookRouter Class` Provides features for the `route`(15.4) handling.

15.4 Classes, Interfaces, Objects and Records

EBrookRoutes Class

Hierarchy

EBrookRoutes > `EBrook`(7.4) > Exception

Description

Handles exceptions for `TBrookRoutes`(15.4).

EBrookRouter Class

Hierarchy

EBrookRouter > `EBrook`(7.4) > Exception

Description

Handles exceptions for `TBrookRouter`(15.4).

TBrookRoute Record

Description

Defines a route item.

Fields

ActionClass `public ActionClass: TBrookActionClass;`

Specifies the class of the action to be called.

Method `public Method: TBrookRequestMethod;`

Specifies a HTTP request `method` of the action to be called.

Pattern `public Pattern: string;`

Specifies the patter of the action to be called.

TBrookRoutes Class

Hierarchy

TBrookRoutes > `TBrookPersistent`(3.4) > TPersistent

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; virtual;`

Description Creates an instance of a `TBrookRoutes`(15.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookRoutes`(15.4) 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.

GetEmptyPatternActionClass

Declaration

```
public procedure GetEmptyPatternActionClass(out AClass:  TBrookActionClass;
out AIndex:  Integer);
```

Description Get the action class with empty pattern.

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 > `TBrookComponent`(3.4) > TComponent

Description

Provides features for the `route`(15.4) 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(AOwner: TComponent); override;`

Description Creates an instance of a `TBrookRouter`(15.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookRouter`(15.4) class.

GetServiceClass

Declaration `public class function GetServiceClass: TBrookRouterClass;`

Description Return the `service`(15.4) class provided by this class.

RegisterService

Declaration `public class procedure RegisterService;`

Description Registers the `service`(15.4) provided by this class.

UnregisterService

Declaration `public class procedure UnregisterService;`

Description Unregisters the `service`(15.4) 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(10.3)` 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:

```
@(longcode
       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`(15.4).

Route

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

Description Runs the `route` processing.

15.5 Types

TBrookRoutesClass

Declaration `TBrookRoutesClass = class of TBrookRoutes;`

Description Is a metaclass for `TBrookRoutes`(15.4) class.

TBrookRouterClass

Declaration `TBrookRouterClass = class of TBrookRouter;`

Description Is a metaclass for `TBrookRouter`(15.4) class.

PBrookRoute

Declaration `PBrookRoute = ^TBrookRoute;`

Description Defines a pointer to the route item.

TBrookMatchPatternEvent

Declaration `TBrookMatchPatternEvent = function(ASender: TObject; APattern, APathInfo: string; out ARedirect: Boolean; out ANames, AValues: TBrookArrayOfString; var AHandled: Boolean): Boolean 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; var AHandled: Boolean) 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; const ANames, AValues: TBrookArrayOfString; 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 16

Unit BrookSession

16.1 Description

Session class.

16.2 Uses

- `BrookClasses`(3)
- `BrookHttpDefs`(10)
- `BrookUtils`(17)
- `BrookException`(7)
- `BrookConsts`(6)
- `Classes`
- `SysUtils`
- `HTTPDefs`
- `DateUtils`

16.3 Overview

`EBrookSession Class` Handles exceptions for `TBrookSession`(16.4).

`TBrookSession Class` Defines features to the session handling.

`TBrookGSession Class` Defines features to the section mapping `field`(16.4) values to object.

16.4 Classes, Interfaces, Objects and Records

EBrookSession Class

Hierarchy

EBrookSession > `EBrook`(7.4) > Exception

Description

Handles exceptions for `TBrookSession`(16.4).

TBrookSession Class

Hierarchy

TBrookSession > `TBrookComponent`(3.4) > TComponent

Description

Defines features to the session 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.

Field `public property Field[constAName:string]: string read GetField write SetField;`

Handles the session `fields`(16.4).

Fields `public property Fields: TStrings read FFields write SetFields;`

The session `fields`.

IgnoredFields

`public property IgnoredFields: TStrings read FIgnoredFields write SetIgnoredFields;`

The ignored `fields`(16.4) by the session.

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 fieds 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 default BROOK_SESS_DEFAULT_TIMEOUT;`

The remaining seconds for the session `finish`(16.4).

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.

AfterStart

`public property AfterStart: TBrookSessionStartEvent read FAfterStart write FAfterStart;`

Is triggered after session `start`(16.4).

BeforeStart

`public property BeforeStart: TBrookSessionStartEvent read FBeforeStart write FBeforeStart;`

Is triggered before session `start`(16.4).

AfterFinish

`public property AfterFinish: TBrookSessionFinishEvent read FAfterFinish write FAfterFinish;`

Is triggered after session `finish`(16.4).

BeforeFinish `public property BeforeFinish: TBrookSessionFinishEvent read FBeforeFinish write FBeforeFinish;`

Is triggered before session `finish`(16.4).

AfterExpire `public property AfterExpire: TBrookSessionExpireEvent read FAfterExpire write FAfterExpire;`

Is triggered after session `expire`(16.4).

BeforeExpire `public property BeforeExpire: TBrookSessionExpireEvent read FBeforeExpire write FBeforeExpire;`

Is triggered before session `expire`(16.4).

Methods

Create

Declaration `public constructor Create(AOwner: TComponent); override;`

Description Creates an instance of a `TBrookSession`(16.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookSession`(16.4) class.

GetFields

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

Description Get an object with the `fields`(16.4) coming from session.

IsExpired

Declaration `public function IsExpired: Boolean; virtual;`

Description Returns `True` if the session has `expired`(16.4).

IsEmpty

Declaration `public function IsEmpty: Boolean; virtual;`

Description Returns `True` if the session fieds is `empty`(16.4).

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`(16.4).

TBrookGSession Class

Hierarchy

TBrookGSession > `TBrookSession`(16.4) > `TBrookComponent`(3.4) > TComponent

Description

Defines features to the section mapping `field`(16.4) values to object.

Properties

Entity `public property Entity: T read FEntity write FEntity;`

Maps `field`(16.4) values to object.

Methods

Create

Declaration `public constructor Create(AOwner: TComponent); override;`

Description Creates an instance of a `TBrookGSession`(16.4) class.

Destroy

Declaration `public destructor Destroy; override;`

Description Frees an instance of `TBrookGSession`(16.4) class.

16.5 Types

TBrookSessionClass

Declaration `TBrookSessionClass = class of TBrookSession;`

Description Is a metaclass for `TBrookSession`(16.4) class.

TBrookSessionStartEvent

Declaration `TBrookSessionStartEvent = procedure(ASender: TObject; ARequest: TBrookRequest; var AHandled: Boolean) of object;`

Description Is a type to the session start event.

PBrookSessionStartEvent

Declaration `PBrookSessionStartEvent = ^TBrookSessionStartEvent;`

Description Defines a pointer to the session start event.

TBrookSessionFinishEvent

Declaration `TBrookSessionFinishEvent = procedure(ASender: TObject; AResponse: TBrookResponse; var AHandled: Boolean) of object;`

Description Is a type to the session finish event.

PBrookSessionFinishEvent

Declaration `PBrookSessionFinishEvent = ^TBrookSessionFinishEvent;`

Description Defines a pointer to the session finish event.

TBrookSessionExpireEvent

Declaration `TBrookSessionExpireEvent = procedure(ASender: TObject; ARequest: TBrookRequest; AResponse: TBrookResponse; var AHandled: Boolean) of object;`

Description Is a type to the session expire event.

PBrookSessionExpireEvent

Declaration `PBrookSessionExpireEvent = ^TBrookSessionExpireEvent;`

Description Defines a pointer to the session expire event.

Chapter 17

Unit BrookUtils

17.1 Description

Utilities unit.

17.2 Uses

- `BrookException`(7)
- `BrookMessages`(13)
- `BrookConsts`(6)
- `BrookHTTPConsts`(9)
- `CustWeb`
- `Classes`
- `SysUtils`
- `TypInfo`

17.3 Overview

`TBrookSettings Record` Stores the Brook settings.

`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.

17.4 Classes, Interfaces, Objects and Records

TBrookSettings Record

Description

Stores the Brook settings.

Fields

Mapped `public Mapped: Boolean;`

Enables the mapping of HTTP request methods.

Charset `public Charset: ShortString;`

Set the default `application`(2.5) `Charset`.

ContentType `public ContentType: ShortString;`

Set the default `application`(2.5) Content-Type.

Page404 `public Page404: string;`

Set the 404 HTML page. The string will be sent as is.

Page404File `public Page404File: string;`

Set the 404 HTML page file. The file content will be sent. This has higher precedence than `TBrookSettings.Page404(17.4)` 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 `TBrookSettings.Page404(17.4)`

Page500 `public Page500: string;`

Set the 500 HTML page. The string will be sent as is.

Page500File `public Page500File: string;`

Set the 500 HTML page file. The file content will be sent. This has higher precedence than `TBrookSettings.Page500(17.4)` 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 `TBrookSettings.Page500(17.4)`

DirectoryForUploads `public DirectoryForUploads: string;`

Set the default directory for uploads.

DeleteUploadedFiles `public DeleteUploadedFiles: Boolean;`

Defines if the temporary uploaded files will be deleted.

KeepUploadedNames `public KeepUploadedNames: Boolean;`

Keeps the original name of the uploaded files.

Configuration `public Configuration: string;`

Set a `configuration` for the `application`(2.5) or for its object members.

RootUrl `public RootUrl: string;`

Set the default root URL. This is used by methods such as `TBrookAction.UrlFor(1.4)`, `TBrookActionHelper.LinkTo`, `TBrookActionHelper.ButtonTo` etc. By default, Brook assumes `SCRIPT_NAME` as root URL.

Port `public Port: Word;`

Set the default `application`(2.5) `port`.

LogActive `public LogActive: Boolean;`

Enables the `application`(2.5) log.

LogFile `public LogFile: TFileName;`

Set a name for the `application`(2.5) log file.

OnError `public OnError: TOnShowRequestException;`

Handles the `application`(2.5) exceptions.

17.5 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:

`BrookGetPathLavel('/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.

17.6 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
```

17.7 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; Configuration: ES; RootUrl: ES; Port: 0; LogActive: False; LogFile:
ES; OnError: nil; );
```

Description Global variable to store Brook settings.