

TSMLSendTo object

[Properties](#)

Unit

[LogBase](#)

Description

The TSMLSendTo object provides the possibility to customize the settings for emailing/url visiting.

You would normally never create an object of this class. This object is created by any logging component automatically.

AttachLog property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property AttachLog: Boolean;

Description

Using this property you can toggle an option to send the error log-file by email or not (send pure message only).

AttachScreenShot property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property AttachScreenShot: Boolean;

Description

Using this property you can toggle an option to send the error screenshot by email or not.

Email property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property EMail: **string**;

Description

Using this property you may customize the recipient email address for bug-report.

Note that you may specify any number of recipients if you'll use delimiter chars (comma or semicolon)

EMailBody property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property EMailBody: **string**;

Description

Using this property you may customize the body of message with bug-report.

By default the standard template used.

EmailOpenBeforeSend property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property EmailOpenBeforeSend: Boolean;

Description

If property is True then generated email message will be opened in edit-mode before sending for any custom modifications by end-user.

Else the generated message added to Outbox folder (in default mailer) automatically.

EmailSubject property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property EmailSubject: **string**;

Description

Using this property you may customize the subject of message with bug-report.

Options property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property Options: [TSMLDialogOptions](#);

Description

You may customize the settings that will apply how error dialog will look.

ResolveNames property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property ResolveNames: Boolean;

Description

If ResolveNames is True then every recipient address will be resolved before sending (optimal send route calculated).
Else the standard smtp-routing used.

NOTE:

better to add the "smtp:" prefix for every recipient address always because some default mailers incorrectly resolve the routing.

ScreenshotFormat property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

property ScreenshotFormat: [TSMLScreenshotFormat](#);

Description

You may specify the format of generated screenshot when error occurred.

WWW property

[See also](#) [Example](#)

Applies to

[TSMLSendTo](#) object

Declaration

```
property WWW: string;
```

Description

You may customize the url/www-link for your homepage.

Active property

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

property Active: Boolean;

Description

Specifies whether or not a logging component is on.

If Active is True then action is logged. Else an action will be not logged.

LogFile property

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
property LogFile: string;
```

Description

LogFile is the name of the file that contains the all logged actions.

Note

File name must include the drive letter and full path. Else default mailer couldn't find the log file for email attachment.

SendTo property

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

property SendTo: [TSMLEndTo](#);

Description

The SendTo property provides the possibility to customize the email settings and options for error dialog.

MAPIErrorDescription method

Applies to

[TSMLogBase](#) component

Declaration

```
function MAPIErrorDescription(intErrorCode: Integer): string;
```

Description

This function allow to get the description of error that returned by default mailer during emailing.

This is internal method used by SendLogByMail method. If error raised, then messagebox displayed with text.

ClearLog method

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
procedure ClearLog;
```

Description

This method allow to clear the current log.

Note that all curent contents will be lost and new file recreated.

SendLogByMail method

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

procedure SendLogByMail;

Description

This method allow to generate the bug-report and send it by email.

Note that all settings from SendTo property will be applied. For example, screenshot generated if required.

SendLogByWWW method

[Example](#)

Applies to

[TSMLogBase](#) component

Declaration

procedure SendLogByWWW;

Description

If you will execute this method then specified homepage will be opened in default browser.

WriteAsDateTime method

[See also](#) [Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
procedure WriteAsDateTime(Value: TDateTime);
```

Description

This method allow to write any custom date-time value to log.

WriteAsFloat method

[See also](#) [Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
procedure WriteAsFloat(Value: Double);
```

Description

This method allow to write any custom numeric value to log.

WriteAsInteger method

[See also](#) [Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
procedure WriteAsInteger(Value: Integer);
```

Description

This method allow to write any custom integer value to log.

WriteAsString method

[See also](#) [Example](#)

Applies to

[TSMLogBase](#) component

Declaration

```
procedure WriteAsString(const Value: string);
```

Description

This method allow to write any custom string value to log.

EMapFileException exception

Unit

[SMMapFile](#)

Description

Defintion for standard exception during map-file processing

TSMMMapFile object

[Properties](#) [Methods](#)

Unit

[SMMapFile](#)

Description

Basic class for map-file parsing/loading.

ExceptAddress property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

property ExceptAddress: DWORD;

Description

This is read-only property that allow to get an information about address for last raised error.

Available in run-time only.

ExceptionAnalyzed property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

property ExceptionAnalyzed: Boolean;

Description

This is read-only property that returns the True if information about last raised error processed successfully.

Available in run-time only.

ExceptLineNumber property

[See also](#)

Applies to

[TSMapFile](#) object

Declaration

property ExceptLineNumber: Integer;

Description

This is read-only property that allow to get an information about line number in sources where last error ocured.

Available in run-time only.

ExceptMethodName property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

property ExceptMethodName: **string**;

Description

This is read-only property that allow to get an information about method/function name in sources where last error occurred.

Available in run-time only.

ExceptUnitName property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

```
property ExceptUnitName: string;
```

Description

This is read-only property that allow to get an information about unit/module name in sources where last error occurred.

Available in run-time only.

MapFileBase property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

property MapFileBase: dword;

Description

This property allows to customize the image base in map-file.

The default value is \$00401000 (0x00401000) but depends from project options (see the Linker page in Project Options dialog for your application)

MapFileName property

[See also](#)

Applies to

[TSMMapFile](#) object

Declaration

```
property MapFileName: string;
```

Description

This property allows to customize the file name where map-file stored.

The default value is yourProject.map but you may customize this value as you like (for example, if you use custom compress method and decompress in temporary file)

Create method

Applies to

[TSMMapFile](#) object

Declaration

constructor Create;

Description

Create is constructs an map-engine and initialize its data before the engine is first used.

LoadExceptionData method

Applies to

[TSMMapFile](#) object

Declaration

```
procedure LoadExceptionData(Address: Pointer);
```

Description

This method will parse the existing map-file and will analyze the loaded map (will find the line number, method name, unit name etc)

TSMExceptionLog component

[See also](#) [Properties](#) [Methods](#) [Events](#) [How to use](#)

Unit

[LogError](#)

Description

This component allow to handle the exceptions occurred in the application.

Every processed error is logged and bug-report could be sent to developer. The extended error dialog displayed and all detailed information is there (error text, memory address, unit name, procedure name, line no etc).

ErrorDialog property

[Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property ErrorDialog: [TSMLErrorDialog](#);

Description

You may customize the dialog style that displayed when error occurred.

FilterClasses property

[See also Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property FilterClasses: [TSMExceptionClasses](#);

Description

You may define the filters for error kind.

Only if occurred error is in specified filter then this error processed by TSMExceptionLog component.

Else standard handler of application will process this error.

FilterGraphics property

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property FilterGraphics: TSMExceptionGraphics;

Description

You may define the filters for graphical error kind.

Only if occurred error is in specified filter then this error processed by TSMExceptionLog component.

Else standard handler of application will process this error.

FilterStandard property

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property FilterStandard: TSMExceptionStandards;

Description

You may define the standard filters for error kind.

Only if occurred error is in specified filter then this error processed by TSMExceptionLog component.

Else standard handler of application will process this error.

Formats property

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property Formats: [TSMLFormats](#);

Description

You may specify the contents of automatical generation in log-file.

MapFile property

Applies to

[TSMExceptionLog](#) component

Declaration

property MapFile: [TSMMapFile](#);

Description

This run-time property points to instance of engine for map-file analyzer

Options property

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property Options: [TSMLAOptions](#);

Description

You may customize the visual style for standard error dialog.

GetActiveControlDetails method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetActiveControlDetails: string;
```

Description

This method generates the string with all information about active control and form (name and caption).

Used internally for logging but you may use same method for your needs

GetApplicationDetails method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetApplicationDetails: string;
```

Description

This method generates the string with all information about application (file name, title, parameters etc).

Used internally for logging but you may use same method for your needs

GetComputerInfo method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetComputerInfo: string;
```

Description

This method collects the information about computer (computer description, logged user etc).

Used internally for logging but you may use same method for your needs

GetFormattedSource method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetFormattedSource: string;
```

Description

This method generates the string with all information about error and analyzed map-file (unit name, procedure, line no etc).

Used internally for logging but you may use same method for your needs

GetMemoryInfo method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetMemoryInfo: string;
```

Description

This method collects the information about memory state (used memory, virtual memory, page file etc).

Used internally for logging but you may use same method for your needs

GetOSInfo method

[See also](#) [Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

```
function GetOSInfo: string;
```

Description

This method collects the information about operating system (Windows version and build).

Used internally for logging but you may use same method for your needs

OnException event

Applies to

[TSMExceptionLog](#) component

Declaration

property OnException: TExceptionEvent;

Description

Using this event you may add any own custom processing for occurred error.

OnSaveException event

[Example](#)

Applies to

[TSMExceptionLog](#) component

Declaration

property OnSaveException: [TSMSaveExceptionEvent](#) ;

Description

This event occurred before logging the error information in bug-report.

Using this event you may skip some errors from logging or add any custom value to standard generated text (will be flushed to file). For example, to add the serial number.

TSMDatasetLog component

[See also](#) [Properties](#) [Methods](#) [Events](#)

Unit

[LogDS](#)

Description

This component allow to handle every action for linked dataset.

Every action could be logged in report-file and sent by email to author.

DataSource property

Applies to

[TSMDatasetLog](#) component

Declaration

property DataSource: TDataSource;

Description

This is a linked DataSource and all actions in this source will be logged.

FilterEvents property

[Example](#)

Applies to

[TSMDatasetLog](#) component

Declaration

property FilterEvents: [TSMldbTypeEvents](#);

Description

You may filter the action kind that must be logged.

OnChange event

[Example](#)

Applies to

[TSMDatasetLog](#) component

Declaration

property OnChange: [TSMLDataChangeEvent](#);

Description

You may specify any custom code that must be executed when logged state changed

EFileNotificationError exception

Unit

[LogFile](#)

Description

Defintion for standard exception for file operations

TSMFileLog component

[See also](#) [Properties](#) [Methods](#) [Events](#)

Unit

[LogFile](#)

Description

This component allow to log the notification for file/directory changes.

Every action could be logged in report-file and sent by email to author.

Folder property

[Example](#)

Applies to

[TSMFileLog](#) component

Declaration

property Folder: **string**;

Description

In this property you may define the drive or directory name that you want to monitor for any changes

Options property

[Example](#)

Applies to

[TSMFileLog](#) component

Declaration

property Options: [TFileNotificationOptions](#);

Description

You may customize the monitor and include only some notification filters for file changes.

TSMNTEventLog component

[See also](#) [Properties](#) [Methods](#)

Unit

[LogNT](#)

Description

This component allow to read the records from Windows Log.

Every action could be logged in report-file and sent by email to author.

EventComputer property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

```
property EventComputer: string;
```

Description

This read-only property returns the computer name for current parsed record of Windows Log

EventCount property

[See also](#)

Applies to

[TSMNTEventLog](#).component

Declaration

property EventCount: dword;

Description

This read-only property returns the number of records in Windows Log

EventID property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventID: DWORD;

Description

This read-only property returns the ID for current parsed record of Windows Log

EventMessageText property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventMessageText: **string**;

Description

This read-only property returns the message of error/warning for current parsed record of Windows Log

EventSID property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventSID: PSID;

Description

This read-only property returns the SID (user ID) for current parsed record of Windows Log

You may convert this SID to username using LookupAccountSID function. See the sample below:

var

UserName: **array**[0..1024] **of** Char;

DomainName: **array**[0..1024] **of** Char;

i, userNameLen, domainNameLen: **Integer**;

Use: SID_NAME_USE;

begin

UserNameLen := SizeOf(UserName);

DomainNameLen := SizeOf(DomainName);

if LookupAccountSID(**nil**, SMNTEventLog.EventSID, UserName, UserNameLen,
DomainName, DomainNameLen, Use) **then**

ShowMessage(UserName)

else

ShowMessage('Unknown user');

end;

EventSource property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventSource: **string**;

Description

This read-only property returns the source name (application or service name) for current parsed record of Windows Log

EventString property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

```
property EventString[Index: dword]: string;
```

Description

This is run-time property that used internally when EventMessageText value generated.

Returns the additional parameters by Index

EventStringCount property

[See also](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventStringCount: DWORD;

Description

This is run-time property that used internally when EventMessageText value generated.

Returns the number of additional parameters

EventTime property

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

```
property EventTime: TDateTime;
```

Description

This read-only property returns the date-time for current parsed record of Windows Log

EventType property

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property EventType: Word;

Description

This read-only property returns the event type for current parsed record of Windows Log

The available types:

elSuccess

elInformation

elWarning

elError

elAuditSuccess

elAuditFailure

Log property

Applies to

[TSMNTEventLog](#) component

Declaration

```
property Log: string;
```

Description

This is the category name for Windows Log.

The default value is Application

Server property

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property Server: **string**;

Description

This property specify the computer name (local or remote) that is a source for Log information. If Server property is empty, this mean that Log from local computer will be parsed.

Source property

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

property Source: **string**;

Description

This property specify the source name (service or application name) that is a source for Log information.

Close method

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

procedure Close;

Description

This method allow to close the opened Log

Open method

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

procedure Open;

Description

This method allow to open the close Log

ReadEvent method

[See also](#) [Example](#)

Applies to

[TSMNTEventLog](#) component

Declaration

procedure ReadEvent(Index: dword);

Description

This method allow to parse the record in opened Log by index

Note that Index is not zero-based. First record have an Index=1

ERegistryNotificationError exception

Unit

[LogRegistry.](#)

Description

Defintion for standard exception for registry operations

TSMRegistryLog component

[See also](#) [Properties](#) [Methods](#) [Events](#)

Unit

[LogRegistry](#).

Description

This component allow to log the notification for Windows registry changes.

Every action could be logged in report-file and sent by email to author.

Key property

[See also](#) [Example](#)

Applies to

[TSMRegistryLog](#) component

Declaration

property Key: `string`;

Description

In this property you may define the key in registry that you want to monitor for any changes

Options property

[Example](#)

Applies to

[TSMRegistryLog](#) component

Declaration

property Options: [TRegistryNotificationOptions](#);

Description

You may customize the monitor and include only some notification filters for registry changes.

RootKey property

[See also](#) [Example](#)

Applies to

[TSMRegistryLog](#) component

Declaration

property RootKey: HKey;

Description

In this property you may define the root key in registry that you want to monitor for any changes

Use RootKey to determine the hierarchy of subkeys an application can access

OnRegistryNotification event

Applies to

[TSMRegistryLog](#) component

Declaration

property OnRegistryNotification: TNotifyEvent;

Description

You may specify any custom code that must be executed when logged state changed

TSMShellLog component

[See also](#) [Properties](#) [Methods](#) [Events](#)

Unit

[LogShell](#)

Description

This component allow to log the notification for Windows Explorer changes.

For example, new file created or network drive mapped or CDROM disk ejected etc

Active property

Applies to

[TSMShellLog](#) component

Declaration

property Active: Boolean;

Description

Specifies whether or not a logging component is on.

If Active is True then action is logged. Else an action will be not logged.

Options property

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property Options: [TSMLNotificationEvents](#);

Description

You may customize the monitor and include only some notification filters for Windows Explorer changes.

Path property

[See also](#) [Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property Path: **string**;

Description

In this property you may define the drive or directory name that you want to monitor for any changes

WatchSubTree property

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property WatchSubTree: Boolean;

Description

Set in True if you want to monitor all sub-directories for specified [Path](#).

OnAssociationChange event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnAssociationChange: TNotifyEvent;

Description

This event fired when any file type association has changed.

OnAttributesChange event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnAttributesChange: [TSMShellNotificationEvent2](#);

Description

This event fired when attributes of an item or folder have changed.

The first paramater is a name of changed item and second must be empty.

OnDriveAdd event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnDriveAdd: [TSMShellNotificationEvent1](#);

Description

This event fired when a drive has been added.

Parameter contains the root of the drive that was added

OnDriveAddGui event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnDriveAddGui: [TSMShellNotificationEvent1](#);

Description

This event fired when a drive has been added and the Shell should create a new window for the drive.

Parameter contains the root of the drive that was added

OnDriveRemove event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnDriveRemove: [TSMShellNotificationEvent1](#);

Description

This event fired when a drive has been removed.

Parameter contains the root of the drive that was removed

OnEndSessionQuery event

Applies to

[TSMShellLog](#) component

Declaration

```
property OnEndSessionQuery: TNotifyEvent;
```

Description

This event fired when Windows ShutDown detected.

OnFileCreate event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFileCreate: [TSMShellNotificationEvent1](#);

Description

This event fired when a new file created in folder.

OnFileDelete event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFileDelete: [TSMShellNotificationEvent1](#);

Description

This event fired when a nonfolder item has been deleted.

Parameter contains the item that was removed

OnFileRename event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFileRename: [TSMShellNotificationEvent2](#);

Description

This event fired when a name of nonfolder item has been changed.

First parameter contains the previous name of item and second parameter contains the new name

OnFolderCreate event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFolderCreate: [TSMShellNotificationEvent1](#);

Description

This event fired when folder has been created.

Parameter contains the folder name that was created

OnFolderRemove event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFolderRemove: [TSMShellNotificationEvent1](#);

Description

This event fired when folder has been removed.

Parameter contains the folder name that was removed

OnFolderRename event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFolderRename: [TSMShellNotificationEvent2](#);

Description

This event fired when a name of folder item has been changed.

First parameter contains the previous name of item and second parameter contains the new name

OnFolderUpdate event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnFolderUpdate: [TSMShellNotificationEvent1](#);

Description

This event fired when contents of an existing folder has been changed, but the folder still exists and has not been renamed.

Parameter contains the folder name that was changed

OnMediaInserted event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnMediaInserted: [TSMShellNotificationEvent1](#);

Description

This event fired when storage media has been inserted into a drive.

Parameter contains the root of the drive that contains the new media

OnMediaRemove event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnMediaRemove: [TSMShellNotificationEvent1](#);

Description

This event fired when storage media has been removed from a drive.

Parameter contains the root of the drive from which the media was removed.

OnNetShare event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnNetShare: [TSMShellNotificationEvent1](#);

Description

This event fired when a folder on the local computer is being shared via a network.

Parameter contains the folder that is being shared.

OnNetUnShare event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnNetUnShare: [TSMShellNotificationEvent1](#);

Description

This event fired when a folder on the local computer is no longer being shared via a network.

Parameter contains the folder that is no longer being shared.

OnServerDisconnect event

[Example](#)

Applies to

[TSMShellLog](#) component

Declaration

property OnServerDisconnect: [TSMShellNotificationEvent1](#);

Description

This event fired when the computer has disconnected from a server

Parameter contains the server from which the computer was disconnected.

TSMLCheckProvider component

[Properties](#) [Methods](#) [Events](#)

Unit

[SMLChecks](#)

Description

This component allow to check in visual style for end-users the any your custom conditions and highlight any control with custom image and hint.

For example, you may check if non-empty value is in editbox before save or to validate the range on numeric value etc

CheckControl property

[See also](#) [Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

property CheckControl: TControl;

Description

This property specify the control that must be checked for valid value and highlighted if error occurred

CheckStatus property

[See also](#) [Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

property CheckStatus: [TSMLCheckStatus](#);

Description

This run-time property returns the state of checked control.

FocusOnError property

[See also](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

property FocusOnError: Boolean;

Description

If FocusOnError is True then checked control (see [CheckControl property](#)) will be focused automatically if incorrect value detected.

HintText property

[Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

```
property HintText: string;
```

Description

Hint contains the text string that can appear when the user moves the mouse pointer over the [linked image](#) (corresponded for error).

If no hint defined, then default text displayed (see [GetHintText property](#))

Picture property

[Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

property Picture: TPicture;

Description

You may customize the picture that will be displayed when error occurred.

If no picture specified, then [default image](#) used.

GetHintText method

[See also](#) [Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

```
function GetHintText: string; virtual;
```

Description

This virtual method allow to customize the default hint text for any inherited check provider.

CreateLinkedImage method

[See also](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

```
procedure CreateLinkedImage(defPicture: TPicture);
```

Description

This method create the image control for [CheckControl](#) and load corresponded picture when error occurred.

Don't call this method directly because will be executed internally.

ExecuteCheck method

[See also](#) [Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

procedure ExecuteCheck;

Description

This method execute the real validation. After execution the [CheckStatus property](#) will be changed.

OnCheck event

[See also](#) [Example](#)

Applies to

[TSMLCheckProvider](#) component

Declaration

property OnCheck: TNotifyEvent;

Description

This event executed after internal validation process. You may add there any own additional checks and change the status if required.

TSMLCheckProviderEdit component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This extended check provider component allow to validate the standard TCustomEdit component (TEdit, TMemo etc) for non-empty text.

Required property

[See also](#) [Example](#)

Applies to

[TSMLCheckProviderEdit](#) component

Declaration

property Required: Boolean;

Description

If Required is True then empty text will raise an error during check.

TSMLCheckProviderNumericEdit component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This extended check provider component allow to validate the standard TCustomEdit component (TEdit, TMaskEdit etc) for numeric value in text.

Also you may specify the range (min/max values) for valid number.

MaxValue property

[See also](#) [Example](#)

Applies to

[TSMLCheckProviderNumericEdit](#) component

Declaration

property MaxValue: Double;

Description

You may define the max value for valid range of control number in editbox.

MinValue property

[See also](#) [Example](#)

Applies to

[TSMLCheckProviderNumericEdit](#) component

Declaration

property MinValue: Double;

Description

You may define the min value for valid range of control number in editbox.

TSMLCheckProviderCheckbox component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This extended check provider component allow to validate the standard TCheckBox component.

If checkbox is not checked, failed status specified. For example, user must confirm that license is accepted before continue.

Checked property

[See also](#) [Example](#)

Applies to

[TSMLCheckProviderCheckbox](#) component

Declaration

property Checked: Boolean;

Description

If Checked is True then corresponded checkbox must be checked. Else failed status occurred during check.

TSMLCheckProviderCombobox component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This extended check provider component allow to validate the standard TCustomCombobox component (TCombobox etc) for non-empty text/selected item.

Required property

[See also](#) [Example](#)

Applies to

[TSMLCheckProviderCombobox](#) component

Declaration

property Required: Boolean;

Description

If Required is True then any item from combobox must be selected. Else failed status will be raised during check.

TSMLCheckProviders component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This collection is the container that keeps all specified check provider instances

Items property

[Example](#)

Applies to

[TSMLCheckProviders](#) component

Declaration

property Items: TSMLCheckProvider;

Description

Items is an index of the items in the collection.

The value of the Index parameter corresponds to the Index property of [TSMLCheckProvider](#). It represents the position of the item in the container.

Processor property

Applies to

[TSMLCheckProviders](#) component

Declaration

property Processor: TComponent;

Description

This readonly property returns in run-time the owner object of [check_processor](#) that keep the collection of checks.

Create method

[See also](#)

Applies to

[TSMLCheckProviders](#) component

Declaration

constructor Create(AProcessor: TComponent);

Description

Creates and initializes a collection.

Add method

[See also](#) [Example](#)

Applies to

[TSMLCheckProviders](#) component

Declaration

```
function Add: TSMLCheckProvider;
```

Description

Creates a new [TSMLCheckProvider](#) instance and adds it to the Items array.

TSMCheckProcessor component

[See also](#) [Properties](#) [Methods](#)

Unit

[SMLChecks](#)

Description

This component allow to check in visual style for end-users the any your custom conditions and highlight any "failed" control with custom image and hint.

For example, you may check if non-empty value is in editbox before save and to validate the range on numeric value etc

BeepOnError property

Applies to

[TSMLCheckProcessor](#) component

Declaration

```
property BeepOnError: Boolean;
```

Description

If BeepOnError property is True then failed check will initiate a standard beep using the computer speaker.

CheckProviders property

[Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

property CheckProviders: [TSMLCheckProviders](#) ;

Description

Use CheckProviders to read or set the check instances. CheckProviders is an indexed collection of [TSMLCheckProvider](#) objects.

Use the properties of the [TSMLCheckProvider](#) objects to specify the attributes of individual checks. Checks can be set at design time through the editor, or programmatically at runtime.

DefaultPicture property

[See also](#) [Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

property DefaultPicture: TPicture;

Description

You may customize the default picture that will be displayed when error occurred (if custom picture is not specified for check provider instance).

If no picture specified, then default image used.

Execute method

[See also](#) [Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

function Execute: Boolean;

Description

This method executes the validation for every item in [check_provider collection](#) and if some failed status found, then corresponded control is highlighted.

FailedCount method

[See also](#) [Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

```
function FailedCount: Integer;
```

Description

This function returns the number of failed check providers.

CheckAllProviders method

[See also](#) [Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

procedure CheckAllProviders;

Description

This method executes the validation for every item in [check_provider collection](#).

HighlightFailed method

[See also](#) [Example](#)

Applies to

[TSMLCheckProcessor](#) component

Declaration

procedure HighlightFailed;

Description

This method allow to hightlight the failed control (to focus a control and create the linked picture with error icon)

TSMAppLog component

[See also](#) [Methods](#) [Events](#)

Unit

[LogApp](#)

Description

This component is a control center for application - message processing, hints, help, idle time, activation/deactivation, minimization/maximization, changing of selected control in any form, form changing, changing of windows settings etc

TSMAppLog component intercepts application-level events.

OnActivate event

[See also](#)

Applies to

[TSMAppLog](#) component

Declaration

property OnActivate: TNotifyEvent;

Description

Occurs when an application becomes active.

Write an OnActivate event handler to perform special processing when the application becomes active.

An application becomes active when it is initially run or when focus moves from another Windows application back to any window of the application.

OnActiveControlChange event

[See also](#) [Example](#)

Applies to

[TSMAppLog](#) component

Declaration

property OnActiveControlChange: TNotifyEvent;

Description

Occurs immediately after input focus changes to a new windowed control.

Write an OnActiveControlChange event handler to take specific action when input focus changes to a new control. The change in focus may be within the active form, or across forms to a new form that then becomes the active form.

When focus moves from one control to another, the following events occur.

- 1 If the new focused control is in a different form, focus moves to the new form.
- 2 Focus moves to the new active control.
- 3 If the active form changed, an OnActiveFormChange event occurs.
- 4 An OnActiveControlChange event occurs.

OnActiveFormChange event

[See also](#) [Example](#)

Applies to

[TSMAppLog](#) component

Declaration

property OnActiveFormChange: TNotifyEvent;

Description

Occurs immediately after a new form becomes active in a multi-form application.

Write an OnActiveFormChange event handler to take specific action when a new form becomes active. OnActiveFormChange occurs when the active form for the application changes, not when a form becomes active because the application becomes active.

OnDeactivate event

[See also](#)

Applies to

[TSMAppLog](#) component

Declaration

property OnDeactivate: TNotifyEvent;

Description

Occurs when an application becomes inactive.

Write an OnDeactivate event handler to perform any special processing that should occur immediately before the application is deactivated. The OnDeactivate event occurs when the user switches from the application to another Windows application.

OnException event

[See also](#)

Applies to

[TSMAAppLog](#) component

Declaration

property OnException: TExceptionEvent;

Description

Occurs when an unhandled exception occurs in the application.

Use OnException to change the default behavior that occurs when an exception is not handled by application code. The OnException event handler is called automatically in the application's HandleException method.

OnException only handles exceptions that occur during message processing. Exceptions that occur before or after the execution of the application's Run method do not generate OnException events.

If an exception passes through the try blocks in the application code, the application automatically calls the HandleException method. Unless the exception object is EAbort, HandleException calls the OnException handler, if one exists. Otherwise, it calls ShowException to display a message box indicating an error occurred.

OnHelp event

Applies to

[TSMAAppLog](#) component

Declaration

property OnHelp: THelpEvent;

Description

Occurs when the application receives a request for help.

Write an OnHelp event handler to perform special processing when the user requests help. The HelpContext and the HelpJump methods automatically trigger the OnHelp event.

Set CallHelp to true to have the application call WinHelp after the event. Set CallHelp to false to prevent the application from calling WinHelp. All application help methods go through OnHelp. The application calls WinHelp only if OnHelp's CallHelp parameter returns True, or if no OnHelp event handler is assigned.

To find the possible values of the Command and Data parameters, search for WinHelp in the Win32 Developer's Reference Help (Win32.HLP) file, which explains the WinHelp API (application programming interface). The possible values for the Data parameter depend upon the value of the Command parameter.

The return value is true if the event handler succeeds, false if it fails.

OnHint event

Applies to

[TSMAppLog](#) component

Declaration

property OnHint: TNotifyEvent;

Description

Occurs when the mouse pointer moves over a control or menu item that can display a Help Hint.

Write an OnHint event handler to perform special processing when the mouse pauses over a control or menu item whose Hint property is not an empty string (").

A common use of the OnHint event is to display the value of a control or menu item's Hint property as the text of a status bar (TStatusBar). The Hint property of a control can specify both a short Help Hint and a (usually) longer hint that appears elsewhere because of code in an OnHint event handler.

OnIdle event

Applies to

[TSMAppLog](#) component

Declaration

property OnIdle: TIdleEvent;

Description

Occurs when an application becomes idle.

Write an OnIdle event handler to perform special processing when an application is idle. An application is idle when it is not processing code. For example, an application is idle when it is waiting for input from the user.

OnIdle is called only once, as the application transitions into an idle state. It is not called continuously unless Done is set to false. Applications that set Done to false consume an inordinate amount of CPU time, which affects overall system performance.

OnMessage event

Applies to

[TSMAAppLog](#) component

Declaration

property OnMessage: TMessageEvent;

Description

Occurs when the application receives a Windows message.

Use OnMessage to trap any or all Windows messages posted to all windows in the application. OnMessage occurs when an application receives a Windows message. OnMessage only receives messages that are posted to the message queue, not those sent directly with the Windows API SendMessage function.

An OnMessage event handler allows an application to respond to messages other than those declared in the events for TApplication. If the application doesn't have a specific handler for an incoming message, the message is dispatched to the window for which it was intended, and Windows handles the message.

Caution:

Thousands of messages per second flow through this event. Be careful when coding the handler, because it can affect the performance of the entire application.

OnMinimize event

Applies to

[TSMAAppLog](#) component

Declaration

property OnMinimize: TNotifyEvent;

Description

Occurs when an application is minimized.

Write an OnMinimize event handler to perform special processing when the application is minimized. The application is minimized, either because the user minimizes the main window, or because of a call to the Minimize method. The application's Icon property determines the icon that represents the minimized application.

OnRestore event

Applies to

[TSMAppLog](#) component

Declaration

property OnRestore: TNotifyEvent;

Description

Occurs when the previously minimized application is restored to its normal size.

Write an OnRestore event handler to perform special processing when the application is restored from the minimized state in which it appears as an icon. An application is restored either because the user restores the application, or because the application calls the Restore method.

Note:

Don't confuse restoring an application with restoring a form or window to its original size. To minimize, maximize, and restore a window or form, change the value of the WindowState property.

OnSettingsChanged event

[Example](#)

Applies to

[TSMAppLog](#) component

Declaration

property OnSettingsChanged: TNotifyEvent;

Description

Occurs when Windows notifies the application that a system-wide setting has changed.

Use OnSettingChange to respond when Windows informs the application that a system-wide setting or policy has changed.

OnShowHint event

Applies to

[TSMAppLog](#) component

Declaration

property OnShowHint: TShowHintEvent;

Description

Occurs when the application is about to display the hint window for a Help Hint.

Write an OnShowHint event handler to change the appearance and behavior of Help Hints.

The HintStr parameter sets the text of the Help Hint. To obtain the text of a hint for a particular control, call the GetLongHint or GetShortHint function, assigning the result to HintStr. To change the text, change the value of this string.

Use the CanShow parameter to permit or prevent the Help Hint from displaying. If CanShow is true, the Help Hint displays. If it is false, the Help Hint does not appear.

Properties

► Run-time only ■ Key properties

- [AttachLog](#)
- [AttachScreenShot](#)
- [EMail](#)
- [EMailBody](#)
- [EMailOpenBeforeSend](#)
- [EMailSubject](#)
- [Options](#)
- [ResolveNames](#)
- [ScreenshotFormat](#)
- [WWW](#)

LogBase unit

In this unit all base types and classed declared.

Components

[TSMLogBase](#)

Objects

[TSMLSendTo](#)

Types

[TSMLDialogOption](#)

[TSMLDialogOptions](#)

[TSMLScreenshotFormat](#)

See also

[AttachScreenShot](#)

[EMail](#)

[EMailBody.](#)

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

AttachLog property example

```
yourTSMExceptionLog.SendTo.AttachLog := False;
```

See also

[AttachLog](#)

[EMail](#)

[EMailBody.](#)

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

AttachScreenShot property example

```
yourTSMExceptionLog.SendTo.AttachScreenShot := False;
```

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

Email property example

```
yourTSMExceptionLog.SendTo.Email :=  
'smtp:support@domain.com;smtp:bugs@domain.com';
```

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

EmailBody property example

```
yourTSMExceptionLog.SendTo.Body := 'Sear Sir/Madam,' + #13#10 + 'I attach generated  
bug-report for your product';
```

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

EmailOpenBeforeSend property example

```
yourTSMExceptionLog.SendTo.EmailOpenBeforeSend := True;
```

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

EmailSubject property example

```
yourTSMExceptionLog.SendTo.EmailSubject := 'MyProduct: bug-report';
```

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[ResolveNames](#)

[ScreenshotFormat](#)

[WWW](#)

Options property example

```
yourTSMExceptionLog.SendTo.Options := [doScreenshot, doEmail, doWeb];
```

TSMLDialogOptions type

[See also](#)

Unit

[LogBase](#)

Declaration

```
type TSMLDialogOptions = set of TSMLDialogOption;
```

Description

This type is a set of supported dialog options.

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ScreenshotFormat](#)

[WWW](#)

ResolveNames property example

```
yourTSMExceptionLog.SendTo.ResolveNames := False;
```


See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[WWW](#)

ScreenshotFormat property example

```
yourTSMExceptionLog.SendTo.ScreenshotFormat := sfJpeg;
```

TSMLScreenshotFormat type

[See also](#)

Unit

[LogBase](#)

Declaration

```
type TSMLScreenshotFormat = (sfBitmap, sfJpeg);
```

Description

This is the enumerated type for available picture formats for generated screenshot:

- **sfBitmap**: to generate screenshot in bitmap format
- **sfJpeg**: to generate screenshot in Jpeg format

See also

[AttachLog](#)

[AttachScreenShot](#)

[EMail](#)

[EMailBody](#).

[EMailOpenBeforeSend](#)

[EMailSubject](#)

[Options](#)

[ResolveNames](#)

[ScreenshotFormat](#)

WWW property example

```
yourTSMExceptionLog.SendTo.WWW := 'http://www.scalabium.com/smlog';
```

Active property example

```
yourTSMExceptionLog.Active := True;  
i := 10 div 0;  
yourTSMExceptionLog.Active := False;
```

TSMLogBase component

[See also](#) [Properties](#) [Methods](#)

Unit

[LogBase](#)

Description

This class is a basic class for all logging components.

LogFile property example

```
yourTSMExceptionLog.LogFile := ExtractFilePath(Application.ExeName) + 'smevents.log';
```


SendTo property example

```
yourTSMExceptionLog.SendTo.WWW := 'http://www.scalabium.com/smlog';  
yourTSMExceptionLog.SendTo.Email := 'smtp:support@scalabium.com';
```

ClearLog method example

```
{ clear log after bug-report send }  
yourTSMExceptionLog.SendLogByEmail;  
yourTSMExceptionLog.ClearLog;
```

SendLogByMail method example

```
yourTSMExceptionLog.SendTo.Email := 'smtp:support@scalabium.com';  
yourTSMExceptionLog.SendTo.AttachScreenshot := True;  
yourTSMExceptionLog.SendLogByEmail;
```

SendLogByWWW method example

```
yourTSMExceptionLog.SendTo.WWW := 'http://www.scalabium.com/smlog';  
yourTSMExceptionLog.SendLogByWWW;
```

See also

[WriteAsFloat](#)

[WriteAsInteger](#)

[WriteAsString](#)

WriteAsDateTime method example

```
dt := EncodeDate(2004, 12, 31);  
yourTSMExceptionLog.WriteAsDateTime(dt);
```

See also

[WriteAsDateTime](#)

[WriteAsInteger](#)

[WriteAsString](#)

WriteAsFloat method example

```
yourTSMExceptionLog.WriteAsFloat(2.35);
```


See also

[WriteAsDateTime](#)

[WriteAsFloat](#)

[WriteAsString](#)

WriteAsInteger method example

```
yourTSMExceptionLog.WriteAsInteger(2138);
```

See also

[WriteAsDateTime](#)

[WriteAsFloat](#)

[WriteAsInteger](#)

WriteAsString method example

```
yourTSMExceptionLog.WriteAsString('my custom mark');
```

SMMapFile unit

In this unit declared the basic engine for map-file parsing.

Objects

[TSMMapFile](#)

Exceptions

[EMapFileException](#)

Properties

▸ Run-time only ▣ Key properties

▸ ▣ [ExceptAddress](#)

▸ ▣ [ExceptionAnalyzed](#)

▸ ▣ [ExceptLineNumber](#)

▸ ▣ [ExceptMethodName](#)

▸ ▣ [ExceptUnitName](#)

▣ [MapFileBase](#)

▣ [MapFileName](#)

Methods

- Key methods

- [Create](#)

~~Destroy~~[{linkDelphi=Destroy_Method}](#)

- [LoadExceptionData](#)

See also

[ExceptionAnalyzed](#)

[ExceptLineNumber](#)

[ExceptMethodName](#)

[ExceptUnitName](#)

See also

[ExceptAddress](#)

[ExceptLineNumber](#)

[ExceptMethodName](#)

[ExceptUnitName](#)

See also

[ExceptAddress](#)

[ExceptionAnalyzed](#)

[ExceptMethodName](#)

[ExceptUnitName](#)

See also

[ExceptAddress](#)

[ExceptionAnalyzed](#)

[ExceptLineNumber](#)

[ExceptUnitName](#)

See also

[ExceptAddress](#)

[ExceptionAnalyzed](#)

[ExceptLineNumber](#)

[ExceptMethodName](#)

See also

[MapFileName](#)

See also

[MapFileBase](#)

See also

[TSMLogBase](#)

Properties

► Run-time only ■ Key properties

■ [ErrorDialog](#)

■ [FilterClasses](#)

■ [FilterGraphics](#)

■ [FilterStandard](#)

■ [Formats](#)

► ■ [MapFile](#)

■ [Options](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [GetActiveControlDetails](#)

- [GetApplicationDetails](#)

- [GetComputerInfo](#)

- [GetFormattedSource](#)

- [GetMemoryInfo](#)

- [GetOSInfo](#)

Events

- Key events
- [OnException](#)
- [OnSaveException](#)

About the TSMExceptionLog component

[TSMExceptionLog reference](#)

Purpose

Just drop this component on main form and you'll receive the useful processing of any errors and exceptions for your application in one place.

Additionally your end-users could automatically send to you the generated bug reports with screenshots in attachment.

Note that you may customize the exception filters if you want to track only some classes of raised errors.

LogError unit

In this unit we will find the definition for component of error logging

Components

[TSMExceptionLog](#)

Types

[TSMExceptionClass](#)

[TSMExceptionClasses](#)

[TSMExceptionGraphic](#)

[TSMExceptionGraphics](#)

[TSMExceptionStandard](#)

[TSMExceptionStandards](#)

[TSMLAOption](#)

[TSMLAOptions](#)

[TSMLErrorDialog](#)

[TSMLFormat](#)

[TSMLFormats](#)

[TSMLSaveExceptionEvent](#)

ErrorDialog property example

```
yourTSMExceptionLog.ErrorDialog := edExtended;
```

TSMLErrorDialog type

Unit

[LogError](#)

Declaration

```
type TSMLErrorDialog = (edNone, edStandard, edExtended);
```

Description

This type enumerates all available kinds for displayed error dialog:

edNone don't display any error dialog (to process in "silence" mode)

edStandard display standard error dialog for application

edExtended display dialog with extended information and support controls

See also

[FilterGraphics](#)

[FilterStandard](#)

FilterClasses property example

{ to process errors occurred during works with streams }

```
SMExceptionLog.FilterClasses := SMExceptionLog.FilterClasses + [efStreamError];
```


TSMExceptionClasses type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionClasses = set of TSMExceptionClass;
```

Description

This type is a set of enumerated classes for error processing.

See also

[FilterClasses](#)

[FilterStandard](#)

FilterGraphics property example

{ to process the errors occurred during graphic image operations }

```
SMExceptionLog.FilterGraphics := SMExceptionLog.FilterGraphics +  
[efInvalidGraphicOperation];
```

See also

[FilterClasses](#)

[FilterGraphics](#)

FilterStandard property example

{ to process all buffer overflow errors }

```
SMExceptionLog.FilterStandard := SMExceptionLog.FilterStandard + [efOverflow];
```

See also

[TSMLFormat](#)

[TSMLFormats](#)

Formats property example

{ to save in log-file the physical address where error occurred }

SMExceptionLog.Formats := SMExceptionLog.Formats + [IfPhysAddress]

TSMLFormats type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMLFormats = set of TSMLFormat;
```

Description

This type is a set of enumerated values for contents generated for every error logging (in bug-report).

See also

[TSMLAOption](#)

[TSMLAOptions](#)

Options property example

{ to show the button for bug-report sending }

```
SMExceptionLog.Options := SMExceptionLog.Options + [laShowSendEMail];
```

TSMLAOptions type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMLAOptions = set of TSMLAOption;
```

Description

This type is a set of enumerated values for visual styles of error dialog.

See also

[GetApplicationDetails](#)

[GetComputerInfo](#)

[GetFormattedSource](#)

[GetMemoryInfo](#)

[GetOSInfo](#)

GetActiveControlDetails method example

```
str := yourSMExceptionLog.GetActiveControlDetails;
```

See also

[GetActiveControlDetails](#)

[GetComputerInfo](#)

[GetFormattedSource](#)

[GetMemoryInfo](#)

[GetOSInfo](#)

GetApplicationDetails method example

```
str := yourSMExceptionLog.GetApplicationDetails;
```

See also

[GetActiveControlDetails](#)

[GetApplicationDetails](#)

[GetFormattedSource](#)

[GetMemoryInfo](#)

[GetOSInfo](#)

GetComputerInfo method example

```
str := yourSMExceptionLog.GetComputerInfo;
```

See also

[GetActiveControlDetails](#)

[GetApplicationDetails](#)

[GetComputerInfo](#)

[GetMemoryInfo](#)

[GetOSInfo](#)

GetFormattedSource method example

```
str := yourSMExceptionLog.GetFormattedSource;
```

See also

[GetActiveControlDetails](#)

[GetApplicationDetails](#)

[GetComputerInfo](#)

[GetFormattedSource](#)

[GetOSInfo](#)

GetMemoryInfo method example

```
str := yourSMExceptionLog.GetMemoryInfo;
```

See also

[GetActiveControlDetails](#)

[GetApplicationDetails](#)

[GetComputerInfo](#)

[GetFormattedSource](#)

[GetMemoryInfo](#)

GetOSInfo method example

```
str := yourSMExceptionLog.GetOSInfo;
```

OnSaveException event example

begin

{ to accept the all errors excepts some Oracle error code}

Accept := (Pos('ORA-20001', E.Message) = 0);

{ to add our custom label to every logged text }

LogText := 'mark label 18AB45' + Chr(13)+Chr(10) + LogText

end;

TSMLSaveExceptionEvent type

Unit

[LogError](#)

Declaration

```
type TSMLSaveExceptionEvent = procedure(Sender: TObject; E:  
Exception; var LogText: string; var Accept: Boolean); of object;
```

Description

TSMLSaveExceptionEvent is a pointer to a method used to notify a logging component an event has occurred.

See also

[TSMLogBase](#)

Properties

▸ Run-time only ▣ Key properties

▣ [DataSource](#)

▣ [FilterEvents](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

Events

- Key events
- [OnChange](#)

LogDS unit

In this unit we will find the definition for component of dataset action logging

Components

[TSMDatasetLog](#)

Types

[TSMLDataChangeEvent](#)

[TSMLDBTypeEvent](#)

[TSMLDBTypeEvents](#)

Constants

defDBEvents

FilterEvents property example

{ to save in log-file the action when current record changed and when value in field changed }

```
SMDatasetLog.FilterEvents := SMDatasetLog.FilterEvents + [dbeRecordChanged,  
dbeUpdateData]
```

TSMLDBTypeEvents type

[See also](#)

Unit

[LogDS](#)

Declaration

```
type TSMLDBTypeEvents = set of TSMLDBTypeEvent;
```

Description

This type is a set of enumerated values for available state of dataset.

OnChange event example

{ to update a text in statusbar }

```
procedure TfrmMain.SMDatasetLogChange(Sender: TObject;  
State: TSMLDBTypeEvent; Field: TField; var LogText: string);  
const  
ASMLDBTypeEvents: array[TSMLDBTypeEvent] of string =  
('ActiveChanged',  
'EditingChanged',  
'DataSetChanged',  
'DataSetScrolled',  
'LayoutChanged',  
'RecordChanged',  
'UpdateData');  
begin  
yourStatusbar.SimpleText := ASMLDBTypeEvents[State];  
end;
```

TSMLDataChangeEvent type

Unit

[LogDS](#)

Declaration

```
type TSMLDataChangeEvent = procedure(Sender: TObject; State:  
TSMLDBTypeEvent; Field: TField; var LogText: string); of object;
```

Description

TSMLDataChangeEvent is a pointer to a method used to notify a logging component an event has occurred.

LogFile unit

In this unit we will find the definition for component of file/directory notifications/logging

Components

[TSMFileLog](#)

Exceptions

[EFileNotificationError](#)

Types

[TFileNotificationOption](#)

[TFileNotificationOptions](#)

[TFileNotifyEvent](#)

See also

[TSMLogBase](#)

Properties

► Run-time only ■ Key properties

■ [Folder](#)

■ [Options](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

Events

- Key events
- [OnFileNotification](#)

Folder property example

```
SMFileLog.Folder := 'C:\DataApp\InBox';
```


Options property example

{ to monitor for changed file sizes including sub-directories }

SMFileLog.Options := SMFileLog.Options + [foSize, foWatchSubFolders]

TFileNotificationOptions type

[See also](#)

Unit

[LogFile](#)

Declaration

```
type TFileNotificationOptions = set of TFileNotificationOption;
```

Description

This type is a set of enumerated values for available file state.

See also

[TSMLogBase](#)

Properties

▸ Run-time only ▣ Key properties

- ▣ [EventComputer](#)
- ▣ [EventCount](#)
- ▣ [EventID](#)
- ▣ [EventMessageText](#)
- ▣ [EventSID](#)
- ▣ [EventSource](#)
- ▣ [EventString](#)
- ▣ [EventStringCount](#)
- ▣ [EventTime](#)
- ▣ [EventType](#)
- ▣ [Log](#)
- ▣ [Server](#)
- ▣ [Source](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [Close](#)

- [Open](#)

- [ReadEvent](#)

LogNT unit

In this unit we will find the definition for component of Windows Log reading

Components

[TSMNTEventLog](#)

Constants

elAuditFailure

elAuditSuccess

elError

elInformation

elSuccess

elWarning

EVENTLOG_AUDIT_FAILURE

EVENTLOG_AUDIT_SUCCESS

EVENTLOG_ERROR_TYPE

EVENTLOG_INFORMATION_TYPE

EVENTLOG_SUCCESS

EVENTLOG_WARNING_TYPE

See also

[EventCount](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[Close](#)

[Open](#)

[ReadEvent](#)

See also

[EventComputer](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSID](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventStringCount](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventTime](#)

[EventType](#)

See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventType](#)

EventTime property example

```
str := FormatDateTime('mmm dd, yyyy tt', SMNTEventLog.EventTime);
```


See also

[EventComputer](#)

[EventID](#)

[EventMessageText](#)

[EventSID](#)

[EventSource](#)

[EventString](#)

[EventStringCount](#)

[EventTime](#)

EventType property example

```
begin  
case SMNTEventLog.EventType of  
  elSuccess: str := 'Success';  
  elInformation: str := 'Information';  
  elWarning: str := 'Warning';  
  elError: str := 'Error';  
  elAuditSuccess: str := 'Audit Success';  
else // elAuditFailure  
  str := 'Audit Failure';  
end;  
end
```

See also

[Source](#)

Server property example

```
SMNTEventLog.Server := "";  
SMNTEventLog.Source := ExtractFileName(Application.ExeName);  
SMNTEventLog.Active := True;
```

See also

[Server](#)

Source property example

```
SMNTEventLog.Server := "";  
SMNTEventLog.Source := ExtractFileName(Application.ExeName);  
SMNTEventLog.Active := True;
```

See also

[Open](#)

Close method example

{ to open a log }

SMNTEventLog.Open;

{ to iterate all records in log }

for i := 1 **to** SMNTEventLog.EventCount **do**

begin

SMNTEventLog.ReadEvent(i);

{ do something }

end;

{ to close a log }

SMNTEventLog.Close;

See also

[Close](#)

Open method example

{ to open a log }

SMNTEventLog.Open;

{ to iterate all records in log }

for i := 1 **to** SMNTEventLog.EventCount **do**

begin

SMNTEventLog.ReadEvent(i);

{ do something }

end;

{ to close a log }

SMNTEventLog.Close;

See also

[EventCount](#)

ReadEvent method example

{ to iterate all records in log }

for i := 1 **to** SMNTEventLog.EventCount **do**

begin

SMNTEventLog.ReadEvent(i);

{ do something }

end;

LogRegistry unit

In this unit we will find the definition for component of Windows registry notifications/logging

Components

[TSMRegistryLog](#)

Exceptions

[ERegistryNotificationError](#)

Types

[TRegistryNotificationOption](#)

[TRegistryNotificationOptions](#)

See also

[TSMLogBase](#)

Properties

► Run-time only ■ Key properties

■ [Key](#).

■ [Options](#)

■ [RootKey](#).

Methods

- Key methods

~~Create~~{linkDelphi>Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

Events

- Key events
- [OnRegistryNotification](#)

See also

[RootKey.](#)

Key property example

```
yourTSMRegistryLog.Key := 'Software\Microsoft';
```

Options property example

{ to monitor the changes in key names }

`yourTSMRegistryLog.Options := yourTSMRegistryLog.Options + [roName]`

TRegistryNotificationOptions type

[See also](#)

Unit

[LogRegistry](#).

Declaration

```
type TRegistryNotificationOptions = set of  
TRegistryNotificationOption;
```

Description

This type is a set of enumerated values for available registry state.

See also

[Key](#).

RootKey property example

```
yourTSMRegistryLog.RootKey := HKEY_LOCAL_MACHINE;
```

See also

[TSMLogBase](#)

Properties

► Run-time only ■ Key properties

■ [Active](#)

■ [Options](#)

■ [Path](#)

■ [WatchSubTree](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

Events

- Key events
- [OnAssociationChange](#)
- [OnAttributesChange](#)
- [OnDriveAdd](#)
- [OnDriveAddGui](#)
- [OnDriveRemove](#)
- [OnEndSessionQuery](#)
- [OnFileCreate](#)
- [OnFileDelete](#)
- [OnFileRename](#)
- [OnFolderCreate](#)
- [OnFolderRemove](#)
- [OnFolderRename](#)
- [OnFolderUpdate](#)
- [OnMediaInserted](#)
- [OnMediaRemove](#)
- [OnNetShare](#)
- [OnNetUnShare](#)
- [OnServerDisconnect](#)

LogShell unit

In this unit we will find the definition for component of Windows Explorer notifications/logging

Components

[TSMSHELLLog](#)

Types

[TSMLNotificationEvent](#)

[TSMLNotificationEvents](#)

[TSMLShellNotificationEvent1](#)

[TSMLShellNotificationEvent2](#)

Constants

SHCNF_ACCEPT_INTERRUPTS

SHCNF_ACCEPT_NON_INTERRUPTS

SHCNF_NO_PROXY

SNM_SHELLNOTIFICATION

Options property example

{ to monitor the when new CDROM inserted in drive }

```
yourTSMSHELLLog.Options := yourTSMSHELLLog.Options + [InMediaInsert]
```

TSMLNotificationEvents type

[See also](#)

Unit

[LogShell](#)

Declaration

```
type TSMLNotificationEvents = set of TSMLNotificationEvent;
```

Description

This type is a set of enumerated values for available states of Windows Explorer.

See also

[WatchSubTree](#)

Path property example

```
yourSMShellLog.Path := 'C:\Data\InBox';  
yourSMShellLog.WatchSubTree := True;  
yourSMShellLog.Active := True;
```


WatchSubTree property example

```
yourSMShellLog.Path := 'C:\Data\InBox';  
yourSMShellLog.WatchSubTree := True;  
yourSMShellLog.Active := True;
```

OnAssociationChange event example

begin

lbLog.Items.Add('File type association change');

end;

OnAttributesChange event example

begin

```
lbLog.Items.Add('Attributes change (from ' + OldValue + ' to ' + NewValue + '');
```

end;

TSMLShellNotificationEvent2 type

Unit

[LogShell](#)

Declaration

```
type TSMLShellNotificationEvent2 = procedure(Sender: TObject;  
OldValue, NewValue: string) of object;
```

Description

You may specify any custom code that must be executed when logged state changed (state with two parameters)

OnDriveAdd event example

begin

lbLog.Items.Add('DriveAdd (' + Value + ')');

end;

TSMLShellNotificationEvent1 type

Unit

[LogShell](#)

Declaration

```
type TSMLShellNotificationEvent1 = procedure(Sender: TObject;  
Value: string) of object;
```

Description

You may specify any custom code that must be executed when logged state changed (state with one parameter)

OnDriveAddGui event example

begin

lbLog.Items.Add('ShellDriveAdd (' + Value + ')');

end;

OnDriveRemove event example

begin

lbLog.Items.Add('DriveRemove (' + Value + ')');

end;

OnFileCreate event example

begin

lbLog.Items.Add('FileCreate (' + Value + ')');

end;

OnFileDelete event example

begin

lbLog.Items.Add('FileDelete (' + Value + ')');

end;

OnFileRename event example

begin

```
lbLog.Items.Add('FileRename (from ' + OldValue + ' to ' + NewValue + ')');
```

end;

OnFolderCreate event example

begin

lbLog.Items.Add('FolderCreate (' + Value + ')');

end;

OnFolderRemove event example

begin

lbLog.Items.Add('FolderRemove (' + Value + ')');

end;

OnFolderRename event example

begin

```
lbLog.Items.Add('FolderRename (from ' + OldValue + ' to ' + NewValue + ');');
```

end;

OnFolderUpdate event example

begin

lbLog.Items.Add('FolderUpdate (' + Value + ')');

end;

OnMediaInserted event example

begin

lbLog.Items.Add('MediaInserted (' + Value + ')');

end;

OnMediaRemove event example

begin

lbLog.Items.Add('MediaRemoved (' + Value + ')');

end;

OnNetShare event example

begin

lbLog.Items.Add('NetShare (' + Value + '');

end;

OnNetUnShare event example

begin

lbLog.Items.Add('NetUnShare (' + Value + ')');

end;

OnServerDisconnect event example

begin

lbLog.Items.Add('ServerDisconnect (' + Value + ');

end;

Properties

► Run-time only ■ Key properties

■ [CheckControl](#)

► ■ [CheckStatus](#)

■ [FocusOnError](#)

■ [HintText](#)

■ [Picture](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [GetHintText](#)

~~Assign~~{linkDelphi=Assign_Method}

- [CreateLinkedImage](#)

- [ExecuteCheck](#)

Events

- Key events
- [OnCheck](#)

SMLChecks unit

In this unit we will find the definition for components that allow to check all end-user actions and highlight the invalid states

Components

[TSMLCheckProvider](#)

[TSMLCheckProviderEdit](#)

[TSMLCheckProviderNumericEdit](#)

[TSMLCheckProviderCheckbox](#)

[TSMLCheckProviderCombobox](#)

[TSMLCheckProviders](#)

[TSMLCheckProcessor](#)

Types

[TSMLCheckStatus](#)

See also

[ExecuteCheck](#)

CheckControl property example

{ add custom check (via event) for date and record kind }

custCheck := TSMLCheckProvider.Create(SMLCheckProcessor.CheckProviders);

custCheck.OnCheck := CustomChecking;

where

procedure TfrmMain.CustomChecking(Sender: TObject);

var

check: TSMLCheckProvider;

begin

check := TSMLCheckProvider(Sender);

check.Picture.Bitmap := **nil**;

{1. user can't define the date for discount less than current date (only for Vendor records)

2. file for import must exist}

if (cbKind.Text = 'Vendor') **and** (dtpFrom.Date < Date()) **then**

begin

check.CheckControl := dtpFrom;

check.HintText := 'Date must be greater than current date';

imgChecks.GetBitmap(1, Check.Picture.Bitmap); {load custom picture}

check.CheckStatus := csFailed

end

else

if (edImport.Text <> '') **and not** FileExists(edImport.Text) **then**

begin

check.CheckControl := edImport;

check.HintText := 'File is not exist

,';

imgChecks.GetBitmap(2, Check.Picture.Bitmap); {load custom picture}

check.CheckStatus := csFailed

end

end;

See also

[ExecuteCheck](#)

[OnCheck](#)

CheckStatus property example

```
if (yourTSMCheckProcessor.CheckProviders[0].CheckStatus = csFailed) then  
  ShowMessage('Please enter valid ZIP code');
```

TSMLCheckStatus type

[See also](#)

Unit

[SMLChecks](#)

Declaration

```
type TSMLCheckStatus = (csUnknown, csValid, csFailed);
```

Description

This type enumerates all available states of checked controls:

csUnknown control is not validated and state is unknown

csValid control validated successfully

csFailed error occurred during control validation

See also

[CheckControl](#)

HintText property example

```
editCheck := TSMLCheckProviderEdit.Create(SMLCheckProcessor.CheckProviders);  
editCheck.CheckControl := edName;  
editCheck.HintText := 'Enter the Name. Value can not be empty';
```

Picture property example

```
{ load custom picture from TImageList }
```

```
imgChecks.GetBitmap(1, Check.Picture.Bitmap);
```

```
check.CheckControl := DateTimePicker1;
```

```
check.HintText := 'Date must be greater than January 01, 2003';
```


See also

[HintText](#)

GetHintText method example

const

```
strDefaultWarningNumericEdit = 'Editbox must contain the value in range between %f and %f';
```

function TSMLCheckProviderNumericEdit.HintMustBeStored: **Boolean**;

begin

```
Result := (HintText = strDefaultWarningNumericEdit);
```

end;

See also

[Picture](#)

See also

[OnCheck](#)

ExecuteCheck method example

```
check.ExecuteCheck;  
if (check.CheckStatus = csFailed) then  
  ShowMessage('validation failed')
```

See also

[ExecuteCheck](#)

OnCheck event example

if Edit1.Text = 'a' **then**

TSMLCheckProvider(Sender).CheckStatus = csValid

See also

[TSMLCheckProvider](#)

Properties

▸ Run-time only ▣ Key properties

▣ [Required](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Assign~~{linkDelphi=Assign_Method}

See also

[CheckControl](#)

Required property example

```
if (check.CheckControl = Edit1) then  
  check.Required := True  
else  
  check.Required := False;
```

See also

[TSMLCheckProvider](#)

[TSMLCheckProviderEdit](#)

Properties

► Run-time only ■ Key properties

■ [MaxValue](#)

■ [MinValue](#)

Methods

■ Key methods

~~Create~~{linkDelphi=Create_Method}

~~GetHintText~~{linkDelphi=GetHintText_Method}

~~Assign~~{linkDelphi=Assign_Method}

See also

[MinValue](#)

MaxValue property example

```
{ check for percent range }  
numCheck.CheckControl := Edit1;  
numCheck.MinValue := 0;  
numCheck.MaxValue := 50;  
numCheck.HintText := 'Valid range for percent is from 0 to 50';
```

See also

[MinValue](#)

MinValue property example

```
{ check for percent range }  
numCheck.CheckControl := Edit1;  
numCheck.MinValue := 0;  
numCheck.MaxValue := 50;  
numCheck.HintText := 'Valid range for percent is from 0 to 50';
```

See also

[TSMLCheckProvider](#)

Properties

► Run-time only ■ Key properties

■ [Checked](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Assign~~{linkDelphi=Assign_Method}

See also

[CheckControl](#)

Checked property example

{ user must accept/confirm changes }

chkCheck := TSMLCheckProviderCheckbox.Create(SMLCheckProcessor.CheckProviders);

chkCheck.CheckControl := CheckBox1;

chkCheck.HintText := 'You must accept changes before save';

See also

[TSMLCheckProvider](#)

Properties

► Run-time only ■ Key properties

■ [Required](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Assign~~{linkDelphi=Assign_Method}

See also

[CheckControl](#)

Required property example

```
{ add the check for combobox }
```

```
cbCheck := TSMLCheckProviderCombobox.Create(SMLCheckProcessor.CheckProviders);
```

```
cbCheck.CheckControl := cbKind;
```

```
cbCheck.HintText := 'Select the Kind from dropdown list';
```

See also

[TSMLCheckProvider](#)

[TSMLCheckProcessor](#)

Properties

▸ Run-time only ▣ Key properties

▸ ▣ [Items](#)

▸ ▣ [Processor](#)

Methods

- Key methods
- [Create](#)
- [Add](#)

Items property example

```
for i := 0 to yourCheckProcessor.CheckProviders.Count-1 do  
begin  
  check := yourCheckProcessor.CheckProviders[i];  
  
  { do something with instanse }  
  check.CheckStatus := csUnknown  
end;
```

See also

[Add](#)

[TSMLCheckProvider](#)

See also

[Create](#)

[Items](#)

[TSMLCheckProvider](#)

Add method example

```
with yourCheckProcessor.CheckProviders.Add do  
begin  
  CheckControl := Edit;  
  OnCheck := MyCheckMethod;  
end;
```

See also

[TSMLCheckProvider](#)

[TSMLCheckProviderEdit](#)

[TSMLCheckProviderNumericEdit](#)

[TSMLCheckProviderCheckbox](#)

[TSMLCheckProviderCombobox](#)

Properties

► Run-time only ■ Key properties

■ [BeepOnError](#)

■ [CheckProviders](#)

■ [DefaultPicture](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [Execute](#)
- [FailedCount](#)
- [CheckAllProviders](#)
- [HighlightFailed](#)

CheckProviders property example

```
for i := 0 to yourCheckProcessor.CheckProviders.Count-1 do  
begin  
  check := yourCheckProcessor.CheckProviders[i];  
  
  { do something with instanse }  
  check.CheckStatus := csUnknown  
end;
```


See also

[Picture](#)

DefaultPicture property example

```
SMLCheckProcessor.DefaultPicture.LoadFromFile('aaa.bmp');
```

See also

[ExecuteCheck](#)

[FailedCount](#)

[CheckAllProviders](#)

[HighlightFailed](#)

Execute method example

```
{check all defined providers}  
if SMLCheckProcessor.Execute then  
begin  
  {close button could be enabled}  
  btnOk.Enabled := True;  
end
```

See also

[ExecuteCheck](#)

[Execute](#)

[CheckAllProviders](#)

[HighlightFailed](#)

FailedCount method example

begin

{check all defined providers}

yourCheckProcessor.CheckAllProviders;

{if failed controls exists, to highlight}

if (yourCheckProcessor.FailedCount > 0) **then**

yourCheckProcessor.HighlightFailed

end;

See also

[ExecuteCheck](#)

[Execute](#)

[FailedCount](#)

[HighlightFailed](#)

CheckAllProviders method example

begin

{check all defined providers}

yourCheckProcessor.CheckAllProviders;

{if failed controls exists, to highlight}

if (yourCheckProcessor.FailedCount > 0) **then**

yourCheckProcessor.HighlightFailed

end;

See also

[ExecuteCheck](#)

[Execute](#)

[CheckAllProviders](#)

[FailedCount](#)

HighlightFailed method example

begin

{check all defined providers}

yourCheckProcessor.CheckAllProviders;

{if failed controls exists, to highlight}

if (yourCheckProcessor.FailedCount > 0) **then**

yourCheckProcessor.HighlightFailed

end;

See also

[TSMLogBase](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

Events

- Key events
- [OnActivate](#)
- [OnActiveControlChange](#)
- [OnActiveFormChange](#)
- [OnDeactivate](#)
- [OnException](#)
- [OnHelp](#)
- [OnHint](#)
- [OnIdle](#)
- [OnMessage](#)
- [OnMinimize](#)
- [OnRestore](#)
- [OnSettingsChanged](#)
- [OnShowHint](#)

LogApp unit

In this unit we will find the definition for component that allow to keep in one place all processing for system messages

Components

[TSMAppLog](#)

See also

[OnDeactivate](#)

See also

[OnActiveFormChange](#)

OnActiveControlChange event example

```
var  
Active: TWinControl;  
i: Integer;  
begin  
Active := Controls[0];  
for i := 0 to ControlCount -1 do  
  begin  
    if Controls[i] is TWinControl then  
      if TWinControl(Controls[i]).Focused then  
        Active := Controls[i];  
      end;  
    if Active <> nil then  
      StatusBar1.SimpleText := GetLongHint(Active.Hint);  
    end;  
  end;  
end;
```

See also

[OnActiveControlChange](#)

OnActiveFormChange event example

begin

Color := clWhite;

Form2.Color := clWhite;

Screen.ActiveForm.Color := clAqua;

end;

See also

[OnActivate](#)

See also

[TSMExceptionLog](#)

OnSettingsChanged event example

```
procedure TForm1.ApplicationEvents1SettingChange(Sender: TObject; Flag: Integer;  
const Section: string; var Result: Integer);  
begin  
Memo1.Lines.BeginUpdate;  
try  
Memo1.Lines.Add(Format('Section = %s', [Section]));  
Memo1.Lines.Add(Format('Flags = %.8x', [Flag]));  
if AnsiSameStr(Section, 'intl') then  
with SysLocale do  
begin  
Memo1.Lines.Add(Format('DefaultLCID = %.8x', [DefaultLCID]));  
Memo1.Lines.Add(Format('PriLangID = %.8x', [PriLangID]));  
Memo1.Lines.Add(Format('SubLangID = %.8x', [SubLangID]));  
  
Memo1.Lines.Add(Format('FarEast = %s', [BoolToStr(FarEast, True)]));  
Memo1.Lines.Add(Format('MiddleEast = %s', [BoolToStr(MiddleEast, True)]));  
end;  
Memo1.Lines.Add("");  
finally  
Memo1.Lines.EndUpdate;  
end;  
end;
```

TSMLDialogOption type

[See also](#)

Unit

[LogBase](#)

Declaration

```
type TSMLDialogOption = (doScreenshot, doEmail, doWeb);
```

Description

This is the enumerated type for available options in error dialog:

- **doScreenshot**: to add the checkbox to dialog where user can enable/disable the screenshot generation
- **doEmail**: to add the EMail button (to send the bug-report by email)
- **doWeb**: to add the HomePage button on dialog

See also

[TSMLDialogOption](#)

[SendTo.Options](#)

See also

[ScreenshotFormat](#)

See also

[TSMExceptionLog](#)

[TSMDatasetLog](#)

[TSMFileLog](#)

[TSMNTEventLog](#)

[TSMRegistryLog](#)

[TSMShellLog](#)

[TSMAppLog](#)

[TSMCheckProcessor](#)

Properties

► Run-time only ■ Key properties

■ [Active](#)

■ [LogFile](#)

■ [SendTo](#)

Methods

- Key methods

~~Create~~{linkDelphi=Create_Method}

~~Destroy~~{linkDelphi=Destroy_Method}

- [MAPIErrorDescription](#)
- [ClearLog](#)
- [SendLogByMail](#)
- [SendLogByWWW](#)
- [WriteAsDateTime](#)
- [WriteAsFloat](#)
- [WriteAsInteger](#)
- [WriteAsString](#)

TSMExceptionClass type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionClass = (efStreamError, efFCreateError,  
efFOpenError, efFilerError, efReadError, efWriteError,  
efClassNotFound, efMethodNotFound, efInvalidImage, efResNotFound,  
efListError, efBitsError, efStringListError, efComponentError,  
efParserError, efOutOfResources, efInvalidOperation, efThread);
```

Description

This type enumerates all available filters for class errors.

See the table below for additional information:

Value VCL-exception class for the failure

efStreamError	EStreamError
efFCreateError	EFCreatError
efFOpenError	EFOpenError
efFilerError	EFilerError
efReadError	EReadError
efWriteError	EWriteError
efClassNotFound	EClassNotFound
efMethodNotFound	EMethodNotFound
efInvalidImage	EInvalidImage
efResNotFound	EResNotFound
efListError	EListError
efBitsError	EBitsError
efStringListError	EStringListError
efComponentError	EComponentError
efParserError	EParseError
efOutOfResources	EOutOfResources
efInvalidOperation	EInvalidOperation
efThread	ETHread

TSMExceptionGraphic type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionGraphic = (efInvalidGraphic,  
efInvalidGraphicOperation);
```

Description

This type enumerates all available filters for graphical errors.

See the table below for additional information:

Value VCL-exception class for the failure

efInvalidGraphic	EInvalidGraphic
efInvalidGraphicOperation	EInvalidGraphicOperation

TSMExceptionGraphics type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionGraphics = set of TSMExceptionGraphic;
```

Description

This type is a set of enumerated values for graphical error processing.

TSMExceptionStandard type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionStandard = (efAbort, efOutOfMemory, efInOutError,  
efIntError, efDivByZero, efRangeError, efIntOverflow, efMathError,  
efInvalidOp, efZeroDivide, efOverflow, efUnderflow,  
efInvalidPointer, efInvalidCast, efConvertError, efAccessViolation,  
efPrivilege, efStackOverflow, efControlC, efVariantError,  
efPropReadOnly, efPropWriteOnly, efExternalException,  
efAssertionFailed, efAbstractError, efIntfCastError,  
efInvalidContainer, efInvalidInsert, efPackageError, efWin32Error);
```

Description

This type enumerates all available filters for standard errors.

See the table below for additional information:

Value VCL-exception class for the failure

efAbort	EAbort
efOutOfMemory	EOutOfMemory
efInOutError	EInOutError
efIntError	EIntError
efDivByZero	EDivByZero
efRangeError	ERangeError
efIntOverflow	EIntOverflow
efMathError	EMathError
efInvalidOp	EInvalidOp
efZeroDivide	EZeroDivide
efOverflow	EOverflow
efUnderflow	EUnderflow
efInvalidPointer	EInvalidPointer
efInvalidCast	EInvalidCast
efConvertError	EConvertError
efAccessViolation	EAccessViolation

efPrivilege EPrivilege
efStackOverflow EStackOverflow
efControlC EControlC
efVariantError EVariantError
efPropReadOnly EPropReadOnly
efPropWriteOnly EPropWriteOnly
efExternalException EExternalException
efAssertionFailed EAssertionFailed
efAbstractError EAbstractError
efIntfCastError EIntfCastError
efInvalidContainer EInvalidContainer
efInvalidInsert EInvalidInsert
efPackageError EPackageError
efWin32Error EWin32Error

TSMExceptionStandards type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMExceptionStandards = set of TSMExceptionStandard;
```

Description

This type is a set of enumerated values for standard error processing.

TSMLAOption type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMLAOption = (laShowSendEMail, laShowSupportLink,  
laShowSourceLine);
```

Description

This type enumerates all available visual styles for standard error dialog:

laShowSendEMail

to add the "Email to author" button in dialog for automatical bug-report sending

laShowSupportLink

to add the "Support homepage" button in dialog for automatical homepage opening

laShowSourceLine

to show the extended information about raised error from map-file

TSMLFormat type

[See also](#)

Unit

[LogError](#)

Declaration

```
type TSMLFormat = (lfDate, lfTime, lfPhysAddress,  
lfExceptionClassName, lfMessage, lfSeparator, lfSourceLine,  
lfApplication, lfActiveControl, lfComputerInfo, lfOSInfo,  
lfMemoryInfo);
```

Description

This type enumerates all available parts of information that could be generated during every error:

lfDate to include the current date when error occurred

lfTime to include the current time when error occurred

lfPhysAddress to include the physical address where error occurred

lfExceptionClassName to include the exception class name

lfMessage to include the error message

lfSeparator to separate every part by system delimiter

lfSourceLine to add the line no where error occurred

lfApplication to generate the extended information about application instance

lfActiveControl to generate the extended information about focused control

lfComputerInfo to collect the extended information about computer

lfOSInfo to collect the extended information about operating system

lfMemoryInfo

to collect the extended information about memory use

See also

[FilterClasses](#)

[TSMExceptionClasse](#)

See also

[Formats](#)

[TSMLFormat](#)

See also

[Options](#)

[TSMLAOption](#)

TSMLDBTypeEvent type

[See also](#)

Unit

[LogDS](#)

Declaration

```
type TSMLDBTypeEvent = (dbeActiveChanged, dbeEditingChanged,  
dbeDataSetChanged, dbeDataSetScrolled, dbeLayoutChanged,  
dbeRecordChanged, dbeUpdateData);
```

Description

This type enumerates all available states of dataset that could be logged:

dbeActiveChanged

respond to changes in the Active property

dbeEditingChanged responds to changes in the editing state of the DataSource

dbeDataSetChanged responds to changes in the dataset

dbeDataSetScrolled allows a response to scrolling the representation of the dataset within the data-aware object

dbeLayoutChanged responds to changes in the representation of the data by the data-aware object

dbeRecordChanged responds to changes in the contents of the current record or field of the dataset

dbeUpdateData provides an interface for writing edits to a record in the dataset

See also

[TSMLDBTypeEvent](#)

[FilterEvents](#)

TFileNotificationOption type

[See also](#)

Unit

[LogFile](#)

Declaration

```
type TFileNotificationOption = (foFile, foFolder, foAttributes,  
foSize, foTime, foWatchSubFolders);
```

Description

This type enumerates all available file states that could be logged:

foFile to monitor the file name change

foFolder to monitor the folder name change

foAttributes to monitor the file attribute change (archive, read-only, hidden, system etc)

foSize to monitor the file size change

foTime to monitor the file date/time change

foWatchSubFolders

to monitor the sub-directories also

TFileNotifyEvent type

Unit

[LogFile](#)

Declaration

```
type TFileNotifyEvent = procedure(Sender: TObject; Action:  
TFileNotificationOption); of object;
```

Description

TFileNotifyEvent is a pointer to a method used to notify a logging component an event has occurred.

OnFileNotification event

[Example](#)

Applies to

[TSMFileLog](#) component

Declaration

property OnFileNotification: [TFileNotifyEvent](#);

Description

You may specify any custom code that must be executed when logged state changed

See also

[Options](#)

[TFileNotificationOption](#)

TRegistryNotificationOption type

[See also](#)

Unit

[LogRegistry](#).

Declaration

```
type TRegistryNotificationOption = (roName, roAttributes,  
roLastSet, roSecurity, roWatchSubFolders);
```

Description

This type enumerates all available registry states that could be logged:

roName to notify about change name (subkey is added or deleted, for example)
roAttributes to notify about change attributes (such as the security descriptor information)
roLastSet to notify about change in value of key
roSecurity to notify about change security descriptor of the key
roWatchSubFolders
to track the changes in subkeys

See also

[Options](#)

[TRegistryNotificationOption](#)

TSMLNotificationEvent type

[See also](#)

Unit

[LogShell](#)

Declaration

```
type TSMLNotificationEvent = (lnAssociationChange,  
lnAttributesChange, lnFileChange, lnFileCreate, lnFileDelete,  
lnFileRename, lnDriveAdd, lnDriveRemove, lnShellDriveAdd,  
lnDriveSpaceChange, lnMediaInsert, lnMediaRemove, lnFolderCreate,  
lnFolderDelete, lnFolderRename, lnFolderUpdate, lnNetShare,  
lnNetUnShare, lnServerDisconnect, lnImageListChange,  
lnEndSessionQuery);
```

Description

This type enumerates all available states of Windows Explorer that could be logged:

lnAssociationChange

A file type association has changed

lnAttributesChange The attributes of an item or folder have changed

lnFileChange A nonfolder item has been changed

lnFileCreate A nonfolder item has been created

lnFileDelete A nonfolder item has been deleted

lnFileRename The name of a nonfolder item has changed

lnDriveAdd A drive has been added.

lnDriveRemove A drive has been removed

lnShellDriveAdd A drive has been added and the Shell should create a new window for the drive

lnDriveSpaceChange The amount of free space on a drive has changed

lnMediaInsert Storage media has been inserted into a drive

lnMediaRemove Storage media has been removed from a drive

lnFolderCreate A folder has been created

lnFolderDelete A folder has been removed

lnFolderRename The name of a folder has changed

lnFolderUpdate The contents of an existing folder have changed,

lnNetShare A folder on the local computer is being shared via the network

lnNetUnShare A folder on the local computer is no longer being shared via the network

lnServerDisconnect The computer has disconnected from a server

lnImageListChange An image in the system image list has changed

lnEndSessionQuery Windows Shutdown detected

See also

[Options](#)

[TSMLNotificationEvent](#)

See also

[CheckStatus](#)

See also

[TSMLDialogOptions](#)

[SendTo.Options](#)

See also

[FilterClasses](#)

[TSMExceptionClasses](#)

See also

[FilterGraphics](#)

[TSMExceptionGraphics](#)

See also

[FilterGraphics](#)

[TSMExceptionGraphic](#)

See also

[FilterStandard](#)

[TSMExceptionStandards](#)

See also

[FilterStandard](#)

[TSMExceptionStandard](#)

See also

[Options](#)

[TSMLAOptions](#)

See also

[Formats](#)

[TSMLFormarts](#)

See also

[TSMLDBTypeEvents](#)

[FilterEvents](#)

See also

[Options](#)

[TFileNotificationOptions](#)

OnFileNotification event example

```
procedure TfrmMain.SMFileLogFileNotification(Sender: TObject;  
Action: TFileNotificationOption);  
const  
arrFileNotification: array[TFileNotificationOption] of string =  
('File', 'Folder', 'Attributes', 'Size', 'Time', 'WatchSubFolders');  
begin  
lbLog.Items.Add('Action: ' + arrFileNotification[Action] + ' changed');  
end;
```

See also

[Options](#)

[TRegistryNotificationOptions](#)

See also

[Options](#)

[TSMLNotificationEvents](#)