

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Modules

Here is a list of all modules:

- **Common type definitions**
- **Application title management**
- **User interface customization.**
- **Support URL management**
- **Server configuration**
- **Report customization**
- **Silent mode configuration**
- **Custom log files management**
- **Custom error handlers**
- **Tracing functions**
- **Internal functions**

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Common type definitions

Classes

struct	BUGTRAP_LOGFILEENTRY_tag
	Log file entry structure return by BT_GetLogFileEntry() for regular log files. More...
struct	BUGTRAP_REGEXPORTENTRY_tag
	Log file entry structure return by BT_GetLogFileEntry() for log files exported from the registry. More...

Typedefs

typedef void(CALLBACK * BT_ErrHandler)(INT_P nErrorHandlerParam)	BT_ErrHandler) Type definition of user-de error handler that's called and after main BugTrap c
typedef enum BUGTRAP_ACTIVITY_tag	BUGTRAP_ACTIVITY Type of action which is performed in response to error.
typedef enum BUGTRAP_DIALOGMESSAGE_tag	BUGTRAP_DIALOGMESSAGE Type of user defined mes displayed on the screen.
typedef enum BUGTRAP_EXITMODE_tag	BUGTRAP_EXITMODE These flags control applic termination mode: what application needs to be terminated after error han control should passed to exception handler.
typedef enum BUGTRAP_FLAGS_tag	BUGTRAP_FLAGS Different BugTrap options: can use any combination of these flags.
typedef enum BUGTRAP_LOGECHOTYPE_tag	BUGTRAP_LOGECHOTYPE Type of log echo mode.
typedef struct BUGTRAP_LOGFILEENTRY_tag	BUGTRAP_LOGFILEENTRY Log file entry structure re BT_GetLogFileEntry() f

	regular log files.
typedef enum BUGTRAP_LOGFLAGS_tag	BUGTRAP_LOGFLAGS Set of log file options.
typedef enum BUGTRAP_LOGFORMAT_tag	BUGTRAP_LOGFORMAT Format of log file.
typedef enum BUGTRAP_LOGLEVEL_tag	BUGTRAP_LOGLEVEL Set of available log levels
typedef enum BUGTRAP_LOGTYPE_tag	BUGTRAP_LOGTYPE Type of log file attached to report.
typedef struct BUGTRAP_REGEXPORTENTRY_tag	BUGTRAP_REGEXPORTENTRY Log file entry structure returned by BT_GetLogFileEntry() function for files exported from the report.
typedef enum BUGTRAP_REPORTFORMAT_tag	BUGTRAP_REPORTFORMAT Format of error report.

Enumerations

enum	BUGTRAP_ACTIVITY_tag { BTA_SHOWUI = 1, BTA_SAVEREPORT = 2, BTA_MAILREPORT = 3, BTA_SENDREPORT = 4 }
	Type of action which is performed in response to the error. More...
enum	BUGTRAP_DIALOGMESSAGE_tag { BTDM_INTRO1 = 1, BTDM_INTRO2 = 2 }
	Type of user defined message displayed on the screen. More...
enum	BUGTRAP_EXITMODE_tag { BTEM_TERMINATEAPP , BTEM_CONTINUESEARCH , BTEM_EXECUTEHANDLER }
	These flags control application termination mode: whatever the application needs to be terminated after error handling, control should passed to another exception handler. More...
enum	BUGTRAP_FLAGS_tag { BTF_NONE = 0x00, BTF_DETAILEDMODE = 0x01, BTF_EDITMAIL = 0x02, BTF_ATTACHREPORT = 0x04, BTF_LISTPROCESSES = 0x08, BTF_SHOWADVANCEDUI = 0x10, BTF_SCREENCAPTURE = 0x20, BTF_INTERCEPTSUEF = 0x80 }
	Different BugTrap options. You can use any combinations of these flags. More...
enum	BUGTRAP_LOGECHOTYPE_tag { BTLE_NONE = 0x00, BTLE_STDOUT = 0x01, BTLE_STDERR = 0x02, BTLE_DBGOUT = 0x04 }
	Type of log echo mode. More...
enum	BUGTRAP_LOGFLAGS_tag { BTLF_NONE = 0x00, BTLF_SHOWLOGLEVEL = 0x01, BTLF_SHOWTIMESTAMP = 0x02 }
	Set of log file options. More...
enum	BUGTRAP_LOGFORMAT_tag { BTLF_XML = 1, BTLF_TEXT = 2, BTLF_STREAM = 3 }

	Format of log file. More...
enum	BUGTRAP_LOGLEVEL_tag { BTLL_NONE = 0, BTLL_ERROR = 1, BTLL_WARNING = 2, BTLL_IMPORTANT = 3, BTLL_INFO = 4, BTLL_VERBOSE = 5, BTLL_ALL = BTLL_INFO }
	Set of available log levels. More...
enum	BUGTRAP_LOGTYPE_tag { BTLT_LOGFILE , BTLT_REGEXPORT }
	Type of log file attached to the report. More...
enum	BUGTRAP_REPORTFORMAT_tag { BTRF_XML = 1, BTRF_TEXT = 2 }
	Format of error report. More...

Typedef Documentation

typedef void(CALLBACK * BT_ErrHandler)(INT_PTR nErrorHandlerPar

Type definition of user-defined error handler that's called before and after main BugTrap dialog.

typedef enum BUGTRAP_ACTIVITY_tag BUGTRAP_ACTIVITY

Type of action which is performed in response to the error.

typedef enum BUGTRAP_DIALOGMESSAGE_tag BUGTRAP_DIALOGMESSAGE

Type of user defined message displayed on the screen.

typedef enum BUGTRAP_EXITMODE_tag BUGTRAP_EXITMODE

These flags control application termination mode: whatever the application needs to be terminated after error handling, control should be passed to another exception handler.

typedef enum BUGTRAP_FLAGS_tag BUGTRAP_FLAGS

Different BugTrap options. You can use any combinations of these flags.

typedef enum BUGTRAP_LOGECHOTYPE_tag BUGTRAP_LOGECHOTYPE

Type of log echo mode.

typedef struct BUGTRAP_LOGFILEENTRY_tag BUGTRAP_LOGFILEENTRY

Log file entry structure return by [BT_GetLogFileEntry\(\)](#) for regular log files.

typedef enum BUGTRAP_LOGFLAGS_tag BUGTRAP_LOGFLAGS

Set of log file options.

`typedef enum BUGTRAP_LOGFORMAT_tag BUGTRAP_LOGFORMAT`

Format of log file.

`typedef enum BUGTRAP_LOGLEVEL_tag BUGTRAP_LOGLEVEL`

Set of available log levels.

`typedef enum BUGTRAP_LOGTYPE_tag BUGTRAP_LOGTYPE`

Type of log file attached to the report.

`typedef struct BUGTRAP_REGEXPORTENTRY_tag BUGTRAP_REGEXPORTENTRY`

Log file entry structure return by [**BT_GetLogFileEntry\(\)**](#) for log files exported from the registry.

`typedef enum BUGTRAP_REPORTFORMAT_tag BUGTRAP_REPORTFORMAT`

Format of error report.

Enumeration Type Documentation

enum BUGTRAP_ACTIVITY_tag

Type of action which is performed in response to the error.

Enumerator:

<i>BTA_SHOWUI</i>	Display BugTrap dialog to allow user selecting desirable option. This is the default option.
<i>BTA_SAVEREPORT</i>	Automatically save error report to file. Use BT_SetReportFilePath() to specify report path.
<i>BTA_MAILREPORT</i>	Automatically send error report by e-mail.
<i>BTA_SENDREPORT</i>	Automatically send bug report to support server.

enum BUGTRAP_DIALOGMESSAGE_tag

Type of user defined message displayed on the screen.

Enumerator:

<i>BTDM_INTRO1</i>	1st line of introduction message.
<i>BTDM_INTRO2</i>	A bunch of lines of introduction message.

enum BUGTRAP_EXITMODE_tag

These flags control application termination mode: whatever the application needs to be terminated after error handling, control should passed to another exception handler.

Enumerator:

<i>BTEM_TERMINATEAPP</i>	BugTrap executes desirable action (usually displays the dialog). The application is being terminated when user closes the dialog. This is default action.
--------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------

BTEM_CONTINUESEARCH BugTrap executes desirable action and continues to search up the stack for a handler.

BTEM_EXECUTEHANDLER BugTrap executes desirable action and transfers control to the exception handlers.

enum **BUGTRAP_FLAGS_tag**

Different BugTrap options. You can use any combinations of these flags.

Enumerator:

<i>BTF_NONE</i>	Equivalent of no options.
<i>BTF_DETAILEDMODE</i>	In detailed mode BugTrap generates mini-dump and packs custom log files within the report.
<i>BTF_EDITMAIL</i>	BugTrap may open its own editor for e-mail messages instead of the editor used by the system. Use this option if you aren't aware of the type of e-mail client installed on user computers.
<i>BTF_ATTACHREPORT</i>	Specify this option to attach bug report to e-mail messages. Be careful with this option. It's potentially dangerous for the detailed mode because of even zipped mini-dump may require huge size. It may exceed the maximum size of e-mail message supported by Internet provider.
<i>BTF_LISTPROCESSES</i>	Set this flag to add list of all processes and loaded modules to the report. Disable this option to speedup report generation.
<i>BTF_SHOWADVANCEDUI</i>	By default BugTrap displays simplified dialog on the screen allowing user to perform only common actions. Enable

	this flag to immediately display dialog with advanced error information.
<i>BTF_SCREENCAPTURE</i>	Bug report in detailed error mode may also include a screen shot automatically captured by BugTrap. By default this option is disabled to minimize report size, but it may be useful if you want to know which dialogs were shown on the screen.
<i>BTF_INTERCEPTSUEF</i>	When enabled, BugTrap injects fake SetUnhandledExceptionFilter() to handle the most severe C runtime errors. Usually such errors are not being reported to custom unhandled exception filters. This flag vanishes any attempts of C runtime library to override unhandled exception filter defined in BugTrap. This option should be used with caution, because such technique may be incompatible with future Windows versions.

enum BUGTRAP_LOGECHOTYPE_tag

Type of log echo mode.

Enumerator:

- BTLE_NONE* Do not duplicate log output to file.
- BTLE_STDOUT* Send a copy of log output to STDOUT (mutually exclusive with STDERR mode).
- BTLE_STDERR* Send a copy of log output to STDERR (mutually exclusive with STDOUT mode).
- BTLE_DBGOUT* Send a copy of log output to the debugger (can accompany STDOUT or STDERR mode).

enum BUGTRAP_LOGFLAGS_tag

Set of log file options.

Enumerator:

<i>BTLF_NONE</i>	Do not show any additional entries in the log.
<i>BTLF_SHOWLOGLEVEL</i>	Use this option if you want to store message levels in a file.
<i>BTLF_SHOWTIMESTAMP</i>	Use this option if you want to store message timestamps in a file. Timestamps are stored in universal (locale independent) format: YYYY/MM/DD HH:MM:SS.

enum BUGTRAP_LOGFORMAT_tag

Format of log file.

Enumerator:

<i>BTLF_XML</i>	Log stored in structured XML format. Log data is entirely stored in memory until application exists or crashes. It is designed for fast IO and it cannot handle large logs.
<i>BTLF_TEXT</i>	Log stored in plain text format. Log data is entirely stored in memory until application exists or crashes. It is designed for fast IO and it cannot handle large logs.
<i>BTLF_STREAM</i>	Log stored in plain text format. Unlike <i>BTLF_XML</i> and <i>BTLF_TEXT</i> log data is not cached in memory. This type of log is optimized for large logs.

enum BUGTRAP_LOGLEVEL_tag

Set of available log levels.

Enumerator:

<i>BTLL_NONE</i>	All message levels are prohibited.
<i>BTLL_ERROR</i>	Error message.
<i>BTLL_WARNING</i>	Warning message.
<i>BTLL_IMPORTANT</i>	Important information message.
<i>BTLL_INFO</i>	Regular information message.
<i>BTLL_VERBOSE</i>	Verbose information message.
<i>BTLL_ALL</i>	All levels of messages are accepted.

enum **BUGTRAP_LOGTYPE_tag**

Type of log file attached to the report.

Enumerator:

<i>BTLT_LOGFILE</i>	Regular log file maintained by the user.
<i>BTLT_REGEXPORT</i>	Log file that's exported from the registry.

enum **BUGTRAP_REPORTFORMAT_tag**

Format of error report.

Enumerator:

<i>BTRF_XML</i>	Report stored in structured XML format.
<i>BTRF_TEXT</i>	Report stored in plain text format.

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Application title management

Functions

BUGTRAP_API LPCTSTR APIENTRY	BT_GetAppName (void) Get application name.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetAppVersion (void) Get application version number.
BUGTRAP_API BOOL APIENTRY	BT_ReadVersionInfo (HMODULE hModule) Read application name and version number from version info block.
BUGTRAP_API void APIENTRY	BT_SetAppName (LPCTSTR pszAppName) Set application name of the project where BugTrap is used.
BUGTRAP_API void APIENTRY	BT_SetAppVersion (LPCTSTR pszAppVersion) Set application version number.

Function Documentation

BUGTRAP_API LPCTSTR APIENTRY BT_GetAppName (void)

Get application name.

BUGTRAP_API LPCTSTR APIENTRY BT_GetAppVersion (void)

Get application version number.

BUGTRAP_API BOOL APIENTRY BT_ReadVersionInfo (HMODULE hModule)

Read application name and version number from version info block.

Note:

hModule can be set to NULL for the main executable.

BUGTRAP_API void APIENTRY BT_SetAppName (LPCTSTR pszAppName)

Set application name of the project where BugTrap is used.

BUGTRAP_API void APIENTRY BT_SetAppVersion (LPCTSTR pszAppVersion)

Set application version number.

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

User interface customization.

Functions

BUGTRAP_API LPCTSTR APIENTRY	BT_GetDialogMessage (BUGTRAP_DIALOGMESSAGE eDialogMessage)
	Get user defined message displayed on the screen.
BUGTRAP_API void APIENTRY	BT_SetDialogMessage (BUGTRAP_DIALOGMESSAGE eDialogMessage, LPCTSTR pszMessage)
	Set user defined message displayed on the screen.

Function Documentation

BUGTRAP_API LPCTSTR APIENTRY BT_GetDialogMessage (BUGTRAP_I

Get user defined message displayed on the screen.

BUGTRAP_API void APIENTRY BT_SetDialogMessage (BUGTRAP_I

LPCTSTR

)

Set user defined message displayed on the screen.

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Support URL management

Functions

BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportEMail (void) Get product support e-mail address.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportURL (void) Get HTTP address of product support site.
BUGTRAP_API void APIENTRY	BT_SetSupportEMail (LPCTSTR pszSupportEMail) Set product support e-mail address.
BUGTRAP_API void APIENTRY	BT_SetSupportURL (LPCTSTR pszSupportURL) Set HTTP address of product support site.

Function Documentation

BUGTRAP_API LPCTSTR APIENTRY BT_GetSupportEMail (void)

Get product support e-mail address.

BUGTRAP_API LPCTSTR APIENTRY BT_GetSupportURL (void)

Get HTTP address of product support site.

BUGTRAP_API void APIENTRY BT_SetSupportEMail (LPCTSTR pszS

Set product support e-mail address.

BUGTRAP_API void APIENTRY BT_SetSupportURL (LPCTSTR pszS

Set HTTP address of product support site.

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Server configuration

Defines

```
#define BUGTRAP_HTTP_PORT 80
```

Use this port for HTTP protocol.

Functions

BUGTRAP_API LPCTSTR APIENTRY	BT_GetNotificationEMail (void)
	Get error notification e-mail address. BugTrap Server may automatically notify product support by e-mail about new bug reports.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportHost (void)
	Get host name of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API SHORT APIENTRY	BT_GetSupportPort (void)
	Get port number of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetNotificationEMail (LPCTSTR pszNotificationEMail)
	Set error notification e-mail address. BugTrap Server may automatically notify product support by e-mail about new bug reports.
BUGTRAP_API void APIENTRY	BT_SetSupportHost (LPCTSTR pszSupportHost)
	Set host name (address) of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetSupportPort (SHORT

	nSupportPort)
	Set port number of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetSupportServer (LPCTSTR pszSupportHost, SHORT nSupportPort) Set host name (address) and port number of BugTrap Server. This server can automatically gather bug reports for the application.

Define Documentation

```
#define BUGTRAP_HTTP_PORT 80
```

Use this port for HTTP protocol.

Function Documentation

BUGTRAP_API LPCTSTR APIENTRY BT_GetNotificationEMail (void)

Get error notification e-mail address. BugTrap Server may automatically notify product support by e-mail about new bug reports.

BUGTRAP_API LPCTSTR APIENTRY BT_GetSupportHost (void)

Get host name of BugTrap Server. This server can automatically gather bug reports for the application.

BUGTRAP_API SHORT APIENTRY BT_GetSupportPort (void)

Get port number of BugTrap Server. This server can automatically gather bug reports for the application.

BUGTRAP_API void APIENTRY BT_SetNotificationEMail (LPCTSTR pszEMail)

Set error notification e-mail address. BugTrap Server may automatically notify product support by e-mail about new bug reports.

BUGTRAP_API void APIENTRY BT_SetSupportHost (LPCTSTR pszHost)

Set host name (address) of BugTrap Server. This server can automatically gather bug reports for the application.

BUGTRAP_API void APIENTRY BT_SetSupportPort (SHORT nSupportPort)

Set port number of BugTrap Server. This server can automatically gather bug reports for the application.

BUGTRAP_API void APIENTRY BT_SetSupportServer (LPCTSTR pszHost, SHORT nSupportPort)

Set host name (address) and port number of BugTrap Server. This

server can automatically gather bug reports for the application.

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Report customization

Defines

```
#define MiniDumpNoDump MAXDWORD
```

Disables mini-dump generation in detailed mode.

Functions

BUGTRAP_API DWORD APIENTRY	BT_GetDumpType (void) Get the type of produced m dump in detailed mode.
BUGTRAP_API DWORD APIENTRY	BT_GetFlags (void) Get current BugTrap option
BUGTRAP_API BUGTRAP_REPORTFORMAT APIENTRY	BT_GetReportFormat (voi Get format of error report.
BUGTRAP_API LPCTSTR APIENTRY	BT GetUserMessage (voic Get user defined message. message may be printed to file.
BUGTRAP_API void APIENTRY	BT_SetDumpType (DWOR dwDumpType) Set the type of produced mi dump in detailed mode.
BUGTRAP_API void APIENTRY	BT_SetFlags (DWORD dw Set various BugTrap option
BUGTRAP_API void APIENTRY	BT_SetReportFormat (BUGTRAP_REPORTFOR eReportFormat) Set format of error report.
BUGTRAP_API void APIENTRY	BT_SetUserMessage (LPCTSTR pszUserMessag Set user defined message. message may be printed to file.
BUGTRAP_API void APIENTRY	BT_SetUserMessageFrom (DWORD dwErrorCode) Set user defined message. message may be printed to

file.

Define Documentation

```
#define MiniDumpNoDump MAXDWORD
```

Disables mini-dump generation in detailed mode.

Function Documentation

BUGTRAP_API DWORD APIENTRY BT_GetDumpType (void)

Get the type of produced mini-dump in detailed mode.

BUGTRAP_API DWORD APIENTRY BT_GetFlags (void)

Get current BugTrap options.

BUGTRAP_API BUGTRAP_REPORTFORMAT APIENTRY BT_GetRep

Get format of error report.

BUGTRAP_API LPCTSTR APIENTRY BT.GetUserMessage (void)

Get user defined message. This message may be printed to a log file.

BUGTRAP_API void APIENTRY BT_SetDumpType (DWORD dwDum

Set the type of produced mini-dump in detailed mode.

BUGTRAP_API void APIENTRY BT_SetFlags (DWORD dwFlags)

Set various BugTrap options.

BUGTRAP_API void APIENTRY BT_SetReportFormat (BUGTRAP_RI

Set format of error report.

BUGTRAP_API void APIENTRY BT_SetUserMessage (LPCTSTR psz

Set user defined message. This message may be printed to a log file.

Referenced by [ATL::BTDialog< BASE_CLASS >::PrivDialogProc\(\)](#),

[MFC::BTWindow< BASE_CLASS >::PrivWindowProc\(\)](#), and

[ATL::BTWindow< BASE_CLASS >::PrivWindowProc\(\)](#).

BUGTRAP_API void APIENTRY BT_SetUserMessageFromCode (DW

Set user defined message. This message may be printed to a log file.

Referenced by [ATL::BTDialog< BASE_CLASS >::PrivDialogProc\(\)](#),
and [ATL::BTWindow< BASE_CLASS >::PrivWindowProc\(\)](#).

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Silent mode configuration

Functions

BUGTRAP_API BUGTRAP_ACTIVITY APIENTRY	BT_GetActivityType (void) Get the type of action which is performed in response to the error.
BUGTRAP_API BUGTRAP_EXITMODE APIENTRY	BT_GetExitMode (void) Get application termination mode.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetMailProfile (void) Get the name of MAPI profile.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetReportFilePath (void) Get the path of error report.
BUGTRAP_API void APIENTRY	BT_SetActivityType (BUGTRAP_ACTIVITY eActivityType) Set the type of action which is performed in response to the error.
BUGTRAP_API void APIENTRY	BT_SetExitMode (BUGTRAP_EXITMODE eExitMode) Set application termination mode.
BUGTRAP_API void APIENTRY	BT_SetMailProfile (LPCTSTR pszMailProfile, LPCTSTR pszMailPassword) Set the name and password of MAPI profile.

BUGTRAP_API void APIENTRY

BT_SetReportFilePath

(LPCTSTR
pszReportFilePath)

Set the path of error report.
This function has effect only
for *BTA_SAVEREPORT*
activity.

Function Documentation

BUGTRAP_API BUGTRAP_ACTIVITY APIENTRY BT_GetActivityType

Get the type of action which is performed in response to the error.

BUGTRAP_API BUGTRAP_EXITMODE APIENTRY BT_GetExitMode (

Get application termination mode.

BUGTRAP_API LPCTSTR APIENTRY BT_GetMailProfile (void)

Get the name of MAPI profile.

BUGTRAP_API LPCTSTR APIENTRY BT_GetReportFilePath (void)

Get the path of error report.

BUGTRAP_API void APIENTRY BT_SetActivityType (BUGTRAP_AC

Set the type of action which is performed in response to the error.

BUGTRAP_API void APIENTRY BT_SetExitMode (BUGTRAP_EXITM

Set application termination mode.

BUGTRAP_API void APIENTRY BT_SetMailProfile (LPCTSTR pszMa

LPCTSTR pszMa

)

Set the name and password of MAPI profile.

BUGTRAP_API void APIENTRY BT_SetReportFilePath (LPCTSTR ps

Set the path of error report. This function has effect only for
BTA_SAVEREPORT activity.

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Custom log files management

Functions

BUGTRAP_API void APIENTRY	BT_AddLogFile (LPCTSTR pszLogFile) Add custom log entry to the list of custom log files attached to bug report.
BUGTRAP_API void APIENTRY	BT_AddRegFile (LPCTSTR pszRegFile, LPCTSTR pszRegKey) Add custom log entry to the list of custom log files attached to bug report.
BUGTRAP_API void APIENTRY	BT_ClearLogFiles (void) Clear the list of custom log files attached to bug report.
BUGTRAP_API void APIENTRY	BT_DeleteLogFile (LPCTSTR pszLogFile) Delete custom log entry from the list of custom log files attached to bug report.
BUGTRAP_API int APIENTRY	BT_ExportRegistryKey (LPCTSTR pszRegFile, LPCTSTR pszRegKey) Export registry key to the file.
BUGTRAP_API DWORD APIENTRY	BT_GetLogFileEntry (INT_PTR nLogFileIndexOrName, BOOL bGetByIndex, BUGTRAP_LOGTYPE *peLogType, PDWORD pdwLogEntrySize, PVOID pLogEntry) This function enumerates log file entries that will be attached

	to the report.
BUGTRAP_API DWORD APIENTRY	BT_GetLogFilesCount (void) Get number of log files attached to the report.

Function Documentation

BUGTRAP_API void APIENTRY BT_AddLogFile (LPCTSTR pszLogFile)

Add custom log entry to the list of custom log files attached to bug report.

**BUGTRAP_API void APIENTRY BT_AddRegFile (LPCTSTR pszRegFile
LPCTSTR pszRegKey)**

Add custom log entry to the list of custom log files attached to bug report.

BUGTRAP_API void APIENTRY BT_ClearLogFiles (void)

Clear the list of custom log files attached to bug report.

BUGTRAP_API void APIENTRY BT_DeleteLogFile (LPCTSTR pszLogFile)

Delete custom log entry from the list of custom log files attached to bug report.

**BUGTRAP_API int APIENTRY BT_ExportRegistryKey (LPCTSTR pszFile
LPCTSTR pszKey)**

Export registry key to the file.

**BUGTRAP_API DWORD APIENTRY BT_GetLogFileEntry (INT_PTR
BOOL
BUGTRAP
PDWORD
PVOID
)**

This function enumerates log file entries that will be attached to the report.

BUGTRAP_API DWORD APIENTRY BT_GetLogFilesCount (void)

Get number of log files attached to the report.

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Custom error handlers

Functions

BUGTRAP_API BT_ErrHandler APIENTRY	BT_GetPostErrorHandler (void) Get address of error handler called after BugTrap dialog.
BUGTRAP_API BT_ErrHandler APIENTRY	BT_GetPreErrorHandler (void) Get address of error handler called before BugTrap dialog.
BUGTRAP_API void APIENTRY	BT_SetPostErrorHandler (BT_ErrHandler pfnPostErrorHandler, INT_PTR nPostErrorHandlerParam) Set address of error handler called after BugTrap dialog.
BUGTRAP_API void APIENTRY	BT_SetPreErrorHandler (BT_ErrHandler pfnPreErrorHandler, INT_PTR nPreErrorHandlerParam) Set address of error handler called before BugTrap dialog.

Function Documentation

BUGTRAP_API BT_ErrHandler APIENTRY BT_GetPostErrHandler (v

Get address of error handler called after BugTrap dialog.

BUGTRAP_API BT_ErrHandler APIENTRY BT_GetPreErrHandler (vo

Get address of error handler called before BugTrap dialog.

BUGTRAP_API void APIENTRY BT_SetPostErrHandler (BT_ErrHand

INT_PTR

)

Set address of error handler called after BugTrap dialog.

BUGTRAP_API void APIENTRY BT_SetPreErrHandler (BT_ErrHand

INT_PTR

)

Set address of error handler called before BugTrap dialog.

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Tracing functions

Functions

BUGTRAP_API BOOL APIENTRY	BT_AppLogEntry (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry)
	Append entry to the end of custom log file. This function is thread safe.
BUGTRAP_API BOOL CDECL	BT_AppLogEntryF (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...)
	Append entry to the end of custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_AppLogEntryV (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList)
	Append entry to the end of custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_ClearLog (INT_PTR iHandle)
	Clear log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_CloseLogFile (INT_PTR iHandle)
	Close custom log file. This function is thread safe.

BUGTRAP_API BOOL APIENTRY	BTFlushLogFile (INT_PTR iHandle) Flush contents of the log file.
BUGTRAP_API DWORD APIENTRY	BTGetLogEchoMode (INT_PTR iHandle) Get echo mode.
BUGTRAP_API LPCTSTR APIENTRY	BTGetLogFileName (INT_PTR iHandle) Get custom log file name. This function is thread safe.
BUGTRAP_API DWORD APIENTRY	BTGetLogFlags (INT_PTR iHandle) Return true if time stamp is added to every log entry.
BUGTRAP_API BUGTRAP_LOGLEVEL APIENTRY	BTGetLogLevel (INT_PTR iHandle) Return minimal log level accepted by tracing functions.
BUGTRAP_API DWORD APIENTRY	BTGetLogSizeInBytes (INT_PTR iHandle) Get maximum log file size in bytes. This function is thread safe.
BUGTRAP_API DWORD APIENTRY	BTGetLogSizeInEntries (INT_PTR iHandle) Get maximum log file size in records. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BTInsLogEntry (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR

	pszEntry)
	Insert entry into the beginning of custom log file. This function is thread safe.
BUGTRAP_API BOOL CDECL	BT_InsLogEntryF (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...)
	Insert entry into the beginning of custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_InsLogEntryV (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList)
	Insert entry into the beginning of custom log file. This function is thread safe.
BUGTRAP_API INT_PTR APIENTRY	BT_OpenLogFile (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)
	Open custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_SetLogEchoMode (INT_PTR iHandle, DWORD dwLogEchoMode)
	Set echo mode.
BUGTRAP_API BOOL APIENTRY	BT_SetLogFlags (INT_PTR iHandle, DWORD dwLogFlags)
	Set true if time stamp is

	added to every log entry.
BUGTRAP_API BOOL APIENTRY	BT_SetLogLevel (INT_PTR iHandle, BUGTRAP_LOGLEVEL eLogLevel) Set minimal log level accepted by tracing functions.
BUGTRAP_API BOOL APIENTRY	BT_SetLogSizeInBytes (INT_PTR iHandle, DWORD dwLogSizeInEntries) Set maximum log file size in bytes. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_SetLogSizeInEntries (INT_PTR iHandle, DWORD dwLogSizeInEntries) Set maximum log file size in records. This function is thread safe.

Function Documentation

```
BUGTRAP_API BOOL APIENTRY BT_AppLogEntry( INT_PTR  
                                         BUGTRAP_LOG  
                                         LPCTSTR  
                                         )
```

Append entry to the end of custom log file. This function is thread safe.
Referenced by [BTTrace::Append\(\)](#).

```
BUGTRAP_API BOOL CDECL BT_AppLogEntryF( INT_PTR  
                                         BUGTRAP_LOG  
                                         LPCTSTR  
                                         )
```

Append entry to the end of custom log file. This function is thread safe.

```
BUGTRAP_API BOOL APIENTRY BT_AppLogEntryV( INT_PTR  
                                         BUGTRAP_LOG  
                                         LPCTSTR  
                                         va_list  
                                         )
```

Append entry to the end of custom log file. This function is thread safe.
Referenced by [BTTrace::AppendF\(\)](#), and [BTTrace::AppendV\(\)](#).

```
BUGTRAP_API BOOL APIENTRY BT_ClearLog( INT_PTR iHandle )
```

Clear log file. This function is thread safe.
Referenced by [BTTrace::Clear\(\)](#).

```
BUGTRAP_API BOOL APIENTRY BT_CloseLogFile( INT_PTR iHandle )
```

Close custom log file. This function is thread safe.

Referenced by [BTTrace::Close\(\)](#).

BUGTRAP_API BOOL APIENTRY BT_FlushLogFile (INT_PTR iHandle)

Flush contents of the log file.

Note:

This function is optional and not required in normal conditions.

Referenced by [BTTrace::Flush\(\)](#).

BUGTRAP_API DWORD APIENTRY BT_GetLogEchoMode (INT_PTR iHandle)

Get echo mode.

Referenced by [BTTrace::GetLogEchoMode\(\)](#).

BUGTRAP_API LPCTSTR APIENTRY BT_GetLogFileName (INT_PTR iHandle)

Get custom log file name. This function is thread safe.

Referenced by [BTTrace::GetFileName\(\)](#).

BUGTRAP_API DWORD APIENTRY BT_GetLogFlags (INT_PTR iHandle)

Return true if time stamp is added to every log entry.

Referenced by [BTTrace::GetLogFlags\(\)](#).

BUGTRAP_API BUGTRAP_LOGLEVEL APIENTRY BT_GetLogLevel (INT_PTR iHandle)

Return minimal log level accepted by tracing functions.

Referenced by [BTTrace::GetLogLevel\(\)](#).

BUGTRAP_API DWORD APIENTRY BT_GetLogSizeInBytes (INT_PTR iHandle)

Get maximum log file size in bytes. This function is thread safe.

Referenced by [BTTrace::GetLogSizeInBytes\(\)](#).

BUGTRAP_API DWORD APIENTRY BT_GetLogSizeInEntries (INT_PTR iHandle)

Get maximum log file size in records. This function is thread safe.

Referenced by [BTTrace::GetLogSizeInEntries\(\)](#).

```
BUGTRAP_API BOOL APIENTRY BT_InsLogEntry ( INT_PTR  
                                             BUGTRAP_LOGI  
                                             LPCTSTR  
                                         )
```

Insert entry into the beginning of custom log file. This function is thread safe.

Referenced by [BTTrace::Insert\(\)](#).

```
BUGTRAP_API BOOL CDECL BT_InsLogEntryF ( INT_PTR  
                                             BUGTRAP_LOGE  
                                             LPCTSTR  
                                         )
```

Insert entry into the beginning of custom log file. This function is thread safe.

```
BUGTRAP_API BOOL APIENTRY BT_InsLogEntryV ( INT_PTR  
                                             BUGTRAP_LOGV  
                                             LPCTSTR  
                                             va_list  
                                         )
```

Insert entry into the beginning of custom log file. This function is thread safe.

Referenced by [BTTrace::InsertF\(\)](#), and [BTTrace::InsertV\(\)](#).

```
BUGTRAP_API INT_PTR APIENTRY BT_OpenLogFile ( LPCTSTR  
                                              BUGTRAP_LOGO  
                                         )
```

Open custom log file. This function is thread safe.

Referenced by [BTTrace::BTTrace\(\)](#), and [BTTrace::Open\(\)](#).

BUGTRAP_API **BOOL APIENTRY BT_SetLogEchoMode (INT_PTR iHandle, DWORD dwLogMode)**

Set echo mode.

Referenced by [BTTrace::SetLogEchoMode\(\)](#).

BUGTRAP_API **BOOL APIENTRY BT_SetLogFlags (INT_PTR iHandle, DWORD dwLogFileFlags)**

Set true if time stamp is added to every log entry.

Referenced by [BTTrace::SetLogFlags\(\)](#).

BUGTRAP_API **BOOL APIENTRY BT_SetLogLevel (INT_PTR iHandle, BUGTRAP_LOG_LEVEL dwLogLevel)**

Set minimal log level accepted by tracing functions.

Referenced by [BTTrace::SetLogLevel\(\)](#).

BUGTRAP_API **BOOL APIENTRY BT_SetLogSizeInBytes (INT_PTR iHandle, DWORD dwLogSizeInBytes)**

Set maximum log file size in bytes. This function is thread safe.

Referenced by [BTTrace::SetLogSizeInBytes\(\)](#).

BUGTRAP_API **BOOL APIENTRY BT_SetLogSizeInEntries (INT_PTR iHandle, DWORD dwLogSizeInEntries)**

Set maximum log file size in records. This function is thread safe.

Referenced by **BTTrace::SetLogSizeInEntries()**.

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

Internal functions

Defines

```
#define BT_SetTerminate()
```

Functions

BUGTRAP_API void CDECL	BT_CallCppFilter (void) Simulates access violation in C++ application. Used by set_terminate().
BUGTRAP_API void CDECL	BT_CallSehFilter (void) Simulates access violation in C++ application. Used by set_terminate().
BUGTRAP_API LONG CALLBACK	BT_CppFilter (PEXCEPTION_POINTERS pExceptionPointers) Executes C++ exception filter.
BUGTRAP_API LPTOP_LEVEL_EXCEPTION_FILTER APIENTRY	BT_InstallSehFilter (void)
BUGTRAP_API void	BT_InterceptSUEF (HMODULE hModule, BOOL bOverride) Explicitly intercepts SetUnhandledExceptionFilter() in a module. This function could be useful for dynamically loaded modules.
BUGTRAP_API BOOL APIENTRY	BT_MailSnapshot (void) Take a snapshot of program memory and e-mail it.
BUGTRAP_API BOOL APIENTRY	BT_MailSnapshotEx (PEXCEPTION_POINTERS pExceptionPointers) Take a snapshot of program memory and e-mail it.

BUGTRAP_API BOOL APIENTRY	BT_SaveSnapshot (PCTSTR pszFileName) Take a snapshot of program memory and save it to a file.
BUGTRAP_API BOOL APIENTRY	BT_SaveSnapshotEx (PEXCEPTION_POINTERS pExceptionPointers, PCTSTR pszFileName) Take a snapshot of program memory and save it to a file.
BUGTRAP_API LONG CALLBACK	BT_SehFilter (PEXCEPTION_POINTERS pExceptionPointers) Executes structured exception filter.
BUGTRAP_API BOOL APIENTRY	BT_SendSnapshot (void) Take a snapshot of program memory and send over network.
BUGTRAP_API BOOL APIENTRY	BT_SendSnapshotEx (PEXCEPTION_POINTERS pExceptionPointers) Take a snapshot of program memory and send over network.
BUGTRAP_API void APIENTRY	BT_UninstallSehFilter (void)

Define Documentation

```
#define BT_SetTerminate( )
```

Installs BugTrap handler to be called by the runtime in response to unhandled C++ exception.

Function Documentation

BUGTRAP_API void CDECL BT_CallCppFilter (void)

Simulates access violation in C++ application. Used by set_terminate().

Note:

Don't call this function directly in your code.

BUGTRAP_API void CDECL BT_CallSehFilter (void)

Simulates access violation in C++ application. Used by set_terminate().

Note:

Don't call this function directly in your code.

BUGTRAP_API LONG CALLBACK BT_CppFilter (PEXCEPTION_POINTEE pException, CONTEXT Context, PEXCEPTION_RECORD ExceptionRecord, PEXCEPTION_MAXIMUM_PARAMETERS NumberOfParameters, PEXCEPTION_MAXIMUM_PARAMETERS *pParameters, PEXCEPTION_ROUTINE Handler, PEXCEPTION_ROUTINE Handler2)

Executes C++ exception filter.

Note:

Don't call this function directly in your code.

Referenced by [ATL::BTDialog< BASE_CLASS >::PrivDialogProc\(\)](#),
[MFC::BTWindow< BASE_CLASS >::PrivWindowProc\(\)](#), and
[ATL::BTWindow< BASE_CLASS >::PrivWindowProc\(\)](#).

BUGTRAP_API LPTOP_LEVEL_EXCEPTION_FILTER APIENTRY BT_InstallExceptionFilter (void)

Explicitly installs BugTrap exception filter.

Note:

Normally you should not call this function.

BUGTRAP_API void BT_InterceptSUEF (HMODULE hModule, BOOL bOverride)

Explicitly intercepts SetUnhandledExceptionFilter() in a module. This function could be useful for dynamically loaded modules.

BUGTRAP_API BOOL APIENTRY BT_MailSnapshot (void)

Take a snapshot of program memory and e-mail it.

BUGTRAP_API BOOL APIENTRY BT_MailSnapshotEx (PEXCEPTION

Take a snapshot of program memory and e-mail it.

BUGTRAP_API BOOL APIENTRY BT_SaveSnapshot (PCTSTR pszFi

Take a snapshot of program memory and save it to a file.

BUGTRAP_API BOOL APIENTRY BT_SaveSnapshotEx (PEXCEPTION

PCTSTR

)

Take a snapshot of program memory and save it to a file.

BUGTRAP_API LONG CALLBACK BT_SehFilter (PEXCEPTION_POI

Executes structured exception filter.

Note:

Don't call this function directly in your code.

BUGTRAP_API BOOL APIENTRY BT_SendSnapshot (void)

Take a snapshot of program memory and send over network.

BUGTRAP_API BOOL APIENTRY BT_SendSnapshotEx (PEXCEPTION

Take a snapshot of program memory and send over network.

BUGTRAP_API void APIENTRY BT_UninstallSehFilter (void)

Explicitly uninstalls BugTrap exception filter and restores the previous exception filter.

Note:

Normally you should not call this function.

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Class List](#)[Class Index](#)[Class Members](#)

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ATL::BTDialog< BASE_CLASS >	
BTTrace	C++ wrapper for tracing API
ATL::BTWindow< BASE_CLASS >	
MFC::BTWindow< BASE_CLASS >	
BUGTRAP_LOGFILEENTRY_tag	Log file entry structure return by BT_GetLogFileEntry() for regular log files
BUGTRAP_REGEXPORTENTRY_tag	Log file entry structure return by BT_GetLogFileEntry() for log files exported from the registry

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Class List](#)[Class Index](#)[Class Members](#)

ATL::BTDialog

ATL::BTDialog< BASE_CLASS > Class Template Reference

List of all members.

Protected Member Functions

T5	BTDialog (T1 param1, T2 param2 , T3 param3 , T4 param4 , T5 param5) _BTDLG_INITIALIZER_(param1)
T4	BTDialog (T1 param1, T2 param2 , T3 param3 , T4 param4) _BTDLG_INITIALIZER_(param1)
T3	BTDialog (T1 param1, T2 param2 , T3 param3) _BTDLG_INITIALIZER_(param1)
template<typename T1 , typename T2 >	BTDialog (T1 param1, T2 param2) _BTDLG_INITIALIZER_(param1) Object initialization (2 parameters).
template<typename T1 >	BTDialog (T1 param1) _BTDLG_INITIALIZER_(param1) Object initialization (1 parameter).
	BTDialog (void) _BTDLG_INITIALIZER_() Object initialization (0 parameters).
virtual DLGPROC	GetDialogProc (void)
	Overridden to return address of custom dialog procedure.

Protected Attributes

T5	param2
T4	param2
T3	param2
T5	param3
T4	param3
T5	param4
	T2
	T3
	T4

Private Member Functions

	LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers) Exception filter.
INT_PTR	PrivDialogProc (HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) This dialog procedure intercepts C++ exceptions.

Private Attributes

DLGPROC	m_pfnBaseDlgProc
	Cached address of base dialog procedure.

Detailed Description

```
template<class BASE_CLASS>
class ATL::BTDialog< BASE_CLASS >
```

This class adds error handling to ATL/WTL dialogs. Only one form of exception handling is permitted per function, therefore this class uses two functions to catch C++ and Windows errors.

Constructor & Destructor Documentation

```
template<class BASE_CLASS>
ATL::BTDialog< BASE_CLASS >::BTDialog ( void ) [protected]
```

Object initialization (0 parameters).

```
template<class BASE_CLASS>
template<typename T1>
ATL::BTDialog< BASE_CLASS >::BTDialog ( T1 param1 ) [explicit,
```

Object initialization (1 parameter).

```
template<class BASE_CLASS>
template<typename T1, typename T2>
ATL::BTDialog< BASE_CLASS >::BTDialog ( T1 param1,
                                T2 param2
) [protected]
```

Object initialization (2 parameters).

```
template<class BASE_CLASS>
T3 ATL::BTDialog< BASE_CLASS >::BTDialog ( T1 param1,
                                T2 param2,
                                T3 param3
) [protecte
```

Type Constraints

```
template<class BASE_CLASS>
T4 ATL::BTDialog< BASE_CLASS >::BTDialog ( T1 param1,
                                T2 param2,
                                T3 param3,
                                T4 param4
) [protecte
```

Type Constraints

```
template<class BASE_CLASS >
T5 ATL::BTDialog< BASE_CLASS >::BTDialog ( T1 param1,
                                T2 param2,
                                T3 param3,
                                T4 param4,
                                T5 param5
) [protect]
```

Type Constraints

Member Function Documentation

```
template<class BASE_CLASS >
```

```
DLGPROC ATL::BTDialog< BASE_CLASS >::GetDialogProc ( void )
```

Overridden to return address of custom dialog procedure.

Returns:

address of dialog procedure.

Type Constraints

```
template<class BASE_CLASS >
```

```
ATL::BTDialog< BASE_CLASS >::LONG ( CALLBACK * m_pfnFilter
```

Exception filter.

```
template<class BASE_CLASS >
```

```
INT_PTR ATL::BTDialog< BASE_CLASS >::PrivDialogProc ( HWND
```

UINT

WPARAM

LPARAM

)

This dialog procedure intercepts C++ exceptions.

Parameters:

hWnd - window handle.

uMsg - specifies the Windows message to be processed.

wParam - provides additional information used in processing the message.

lParam - provides additional information used in processing the message.

Returns:

the return value depends on the message.

References [BT_CppFilter\(\)](#), [BT_SetUserMessage\(\)](#),
[BT_SetUserMessageFromCode\(\)](#), and [ATL::BTDialog<](#)

BASE_CLASS >::m_pfnBaseDlgProc.

Member Data Documentation

template<class BASE_CLASS >
DLGPROC ATL::BTDialog< BASE_CLASS >::m_pfnBaseDlgProc [pr]

Cached address of base dialog procedure.

Referenced by **ATL::BTDialog< BASE_CLASS >::PrivDialogProc()**.

template<class BASE_CLASS >
T5 ATL::BTDialog< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >
T4 ATL::BTDialog< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >
T3 ATL::BTDialog< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >
T5 ATL::BTDialog< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >
T4 ATL::BTDialog< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >
T5 ATL::BTDialog< BASE_CLASS >::param4 [protected]

template<class BASE_CLASS >
ATL::BTDialog< BASE_CLASS >::T2 [protected]

template<class BASE_CLASS >
ATL::BTDialog< BASE_CLASS >::T3 [protected]

template<class BASE_CLASS >

ATL::BTDialog< BASE_CLASS >::T4 [protected]

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BTTrace Class Reference

C++ wrapper for tracing API. More...

[List of all members.](#)

Public Member Functions

BOOL	Append (LPCTSTR pszEntry) const Append entry to the end of custom log file.
BOOL	Append (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry) const Append entry to the end of custom log file.
BOOL	AppendF (LPCTSTR pszFormat,...) const Append entry to the end of custom log file.
BOOL	AppendF (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...) const Append entry to the end of custom log file.
BOOL	AppendV (LPCTSTR pszFormat, va_list argList) const Append entry to the end of custom log file.
BOOL	AppendV (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList) const Append entry to the end of custom log file.
void	Attach (INT_PTR iHandle) Attach log file handle.
	BTTrace (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat) Initialize the object.
	BTTrace (INT_PTR iHandle) Initialize the object.
	BTTrace (void) Initialize the object.
BOOL	Clear (void) const Clear log file.
BOOL	Close (void)

		Close log file.
	void	Detach (void) Detach log file handle.
	BOOL	Flush (void) const Flush contents of the log file.
BUGTRAP_LOGLEVEL		GetDefaultLogLevel (void) const Get default log level.
	LPCTSTR	GetFileName (void) const Get custom log file name.
	INT_PTR	GetHandle (void) const Get custom log file handle.
	DWORD	GetLogEchoMode (void) const Get log echo mode.
	DWORD	GetLogFlags (void) const Return current set of log flags.
BUGTRAP_LOGLEVEL		GetLogLevel (void) const Return minimal log level accepted by tracing functions.
	DWORD	GetLogSizeInBytes (void) const Get custom log file size in bytes.
	DWORD	GetLogSizeInEntries (void) const Get custom log file size in records.
	BOOL	Insert (LPCTSTR pszEntry) const Insert entry into the beginning of custom log file.
	BOOL	Insert (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry) const Insert entry into the beginning of custom log file.
	BOOL	InsertF (LPCTSTR pszFormat,...) const Insert entry into the beginning of custom log file.

	BOOL	InsertF (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...) const
		Insert entry into the beginning of custom log file.
	BOOL	InsertV (LPCTSTR pszFormat, va_list argList) const
		Insert entry into the beginning of custom log file.
	BOOL	InsertV (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList) const
		Insert entry into the beginning of custom log file.
	INT_PTR	Open (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)
		Open log file.
	void	SetDefaultLogLevel (BUGTRAP_LOGLEVEL eDefaultLogLevel)
		Set default log level.
	BOOL	SetLogEchoMode (DWORD dwLogEchoMode) const
		Set log echo mode.
	BOOL	SetLogFlags (DWORD dwLogFlags) const
		Set new set of log flags.
	BOOL	SetLogLevel (BUGTRAP_LOGLEVEL eLogLevel) const
		Set minimal log level accepted by tracing functions.
	BOOL	SetLogSizeInBytes (DWORD dwSize) const
		Set custom log file size in bytes.
	BOOL	SetLogSizeInEntries (DWORD dwSize) const
		Set custom log file size in records.

-BTTrace (void)

Destroy the object.

Private Member Functions

	BTTrace (const BTTrace &rTrace) Prevent object from being accidentally copied.
BTTrace &	operator= (const BTTrace &rTrace) Prevent object from being accidentally copied.

Private Attributes

BUGTRAP_LOGLEVEL	m_eDefaultLogLevel
	Default log level.
INT_PTR	m_iHandle
	Log file handle.

Detailed Description

C++ wrapper for tracing API.

Constructor & Destructor Documentation

BTTrace::BTTrace (void)

Initialize the object.

References [Detach\(\)](#).

BTTrace::BTTrace (INT_PTR iHandle)

Initialize the object.

References [BTLL_INFO](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

BTTrace::BTTrace (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)

Initialize the object.

References [BT_OpenLogFile\(\)](#), [BTLL_INFO](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

BTTrace::~BTTrace (void)

Destroy the object.

References [Close\(\)](#).

BTTrace::BTTrace (const BTTrace & rTrace) [private]

Prevent object from being accidentally copied.

Member Function Documentation

BOOL BTTrace::Append (LPCTSTR pszEntry) const

Append entry to the end of custom log file.

References [BT_AppLogEntry\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

**BOOL BTTrace::Append (BUGTRAP_LOGLEVEL eLogLevel,
LPCTSTR pszEntry
)** **const**

Append entry to the end of custom log file.

References [BT_AppLogEntry\(\)](#), and [m_iHandle](#).

**BOOL BTTrace::AppendF (LPCTSTR pszFormat,
...
)** **const**

Append entry to the end of custom log file.

References [BT_AppLogEntryV\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

**BOOL BTTrace::AppendF (BUGTRAP_LOGLEVEL eLogLevel,
LPCTSTR pszFormat,
...
)** **const**

Append entry to the end of custom log file.

References [BT_AppLogEntryV\(\)](#), and [m_iHandle](#).

**BOOL BTTrace::AppendV (LPCTSTR pszFormat,
va_list argList
)** **const**

Append entry to the end of custom log file.

References [BT_AppLogEntryV\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

```
BOOL BTTrace::AppendV( BUGTRAP_LOGLEVEL eLogLevel,  
                      LPCTSTR             pszFormat,  
                      va_list              argList  
                      )  
const
```

Append entry to the end of custom log file.

References [BT_AppLogEntryV\(\)](#), and [m_iHandle](#).

```
void BTTrace::Attach( INT_PTR iHandle )
```

Attach log file handle.

References [m_iHandle](#).

```
BOOL BTTrace::Clear( void ) const
```

Clear log file.

References [BT_ClearLog\(\)](#), and [m_iHandle](#).

```
BOOL BTTrace::Close( void )
```

Close log file.

References [BT_CloseLogFile\(\)](#), [Detach\(\)](#), and [m_iHandle](#).

Referenced by [Open\(\)](#), and [~BTTrace\(\)](#).

```
void BTTrace::Detach( void )
```

Detach log file handle.

References [BTLL_INFO](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

Referenced by [BTTrace\(\)](#), and [Close\(\)](#).

BOOL BTTrace::Flush (void) const

Flush contents of the log file.

References [BTFlushLogFile\(\)](#), and [m_iHandle](#).

BUGTRAP_LOGLEVEL BTTrace::GetDefaultLogLevel (void) const

Get default log level.

References [m_eDefaultLogLevel](#).

LPCTSTR BTTrace::GetFileName (void) const

Get custom log file name.

References [BTGetLogFileName\(\)](#), and [m_iHandle](#).

INT_PTR BTTrace::GetHandle (void) const

Get custom log file handle.

References [m_iHandle](#).

DWORD BTTrace::GetLogEchoMode (void) const

Get log echo mode.

References [BTGetLogEchoMode\(\)](#), and [m_iHandle](#).

DWORD BTTrace::GetLogFlags (void) const

Return current set of log flags.

References [BTGetLogFlags\(\)](#), and [m_iHandle](#).

BUGTRAP_LOGLEVEL BTTrace::GetLogLevel (void) const

Return minimal log level accepted by tracing functions.

References [BTGetLogLevel\(\)](#), and [m_iHandle](#).

DWORD BTTrace::GetLogSizeInBytes (void) const

Get custom log file size in bytes.

References [BT_GetLogSizeInBytes\(\)](#), and [m_iHandle](#).

DWORD BTTrace::GetLogSizeInEntries (void) const

Get custom log file size in records.

References [BT_GetLogSizeInEntries\(\)](#), and [m_iHandle](#).

BOOL BTTrace::Insert (LPCTSTR pszEntry) const

Insert entry into the beginning of custom log file.

References [BT_InsLogEntry\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

BOOL BTTrace::Insert (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry) const

Insert entry into the beginning of custom log file.

References [BT_InsLogEntry\(\)](#), and [m_iHandle](#).

BOOL BTTrace::InsertF (LPCTSTR pszFormat, ...) const

Insert entry into the beginning of custom log file.

References [BT_InsLogEntryV\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

BOOL BTTrace::InsertF (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, ...) const

Insert entry into the beginning of custom log file.

References [BT_InsLogEntryV\(\)](#), and [m_iHandle](#).

```
BOOL BTTrace::InsertV ( LPCTSTR pszFormat,  
                        va_list argList  
                    ) const
```

Insert entry into the beginning of custom log file.

References [BT_InsLogEntryV\(\)](#), [m_eDefaultLogLevel](#), and [m_iHandle](#).

```
BOOL BTTrace::InsertV ( BUGTRAP_LOGLEVEL eLogLevel,  
                        LPCTSTR pszFormat,  
                        va_list argList  
                    ) const
```

Insert entry into the beginning of custom log file.

References [BT_InsLogEntryV\(\)](#), and [m_iHandle](#).

```
INT_PTR BTTrace::Open ( LPCTSTR pszLogFileName  
                        BUGTRAP_LOGFORMAT eLogFormat  
                    )
```

Open log file.

References [BT_OpenLogFile\(\)](#), [Close\(\)](#), and [m_iHandle](#).

```
BTTrace& BTTrace::operator= ( const BTTrace & rTrace ) [private]
```

Prevent object from being accidentally copied.

```
void BTTrace::SetDefaultLogLevel ( BUGTRAP_LOGLEVEL eDefaultLevel )
```

Set default log level.

References [m_eDefaultLogLevel](#).

```
BOOL BTTrace::SetLogEchoMode ( DWORD dwLogEchoMode ) const
```

Set log echo mode.

References [BT_SetLogEchoMode\(\)](#), and [m_iHandle](#).

BOOL BTTrace::SetLogFlags (DWORD dwLogFlags) const

Set new set of log flags.

References [BT_SetLogFlags\(\)](#), and [m_iHandle](#).

BOOL BTTrace::SetLogLevel (BUGTRAP_LOGLEVEL eLogLevel) const

Set minimal log level accepted by tracing functions.

References [BT_SetLogLevel\(\)](#), and [m_iHandle](#).

BOOL BTTrace::SetLogSizeInBytes (DWORD dwSize) const

Set custom log file size in bytes.

References [BT_SetLogSizeInBytes\(\)](#), and [m_iHandle](#).

BOOL BTTrace::SetLogSizeInEntries (DWORD dwSize) const

Set custom log file size in records.

References [BT_SetLogSizeInEntries\(\)](#), and [m_iHandle](#).

Member Data Documentation

BUGTRAP_LOGLEVEL BTTrace::m_eDefaultLogLevel [private]

Default log level.

Referenced by [Append\(\)](#), [AppendF\(\)](#), [AppendV\(\)](#), [BTTrace\(\)](#), [Detach\(\)](#), [GetDefaultLogLevel\(\)](#), [Insert\(\)](#), [InsertF\(\)](#), [InsertV\(\)](#), and [SetDefaultLogLevel\(\)](#).

INT_PTR BTTrace::m_iHandle [private]

Log file handle.

Referenced by [Append\(\)](#), [AppendF\(\)](#), [AppendV\(\)](#), [Attach\(\)](#), [BTTrace\(\)](#), [Clear\(\)](#), [Close\(\)](#), [Detach\(\)](#), [Flush\(\)](#), [GetFileName\(\)](#), [GetHandle\(\)](#), [GetLogEchoMode\(\)](#), [GetLogFlags\(\)](#), [GetLogLevel\(\)](#), [GetLogSizeInBytes\(\)](#), [GetLogSizeInEntries\(\)](#), [Insert\(\)](#), [InsertF\(\)](#), [InsertV\(\)](#), [Open\(\)](#), [SetLogEchoMode\(\)](#), [SetLogFlags\(\)](#), [SetLogLevel\(\)](#), [SetLogSizeInBytes\(\)](#), and [SetLogSizeInEntries\(\)](#).

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Class List](#)[Class Index](#)[Class Members](#)

ATL::BTWindow

ATL::BTWindow< BASE_CLASS > Class Template Reference

List of all members.

Protected Member Functions

T5	BTWindow (T1 param1, T2 param2 , T3 param3 , T4 param4 , T5 param5) _BTWND_INITIALIZER_(param1)
T4	BTWindow (T1 param1, T2 param2 , T3 param3 , T4 param4) _BTWND_INITIALIZER_(param1)
template<typename T1 , typename T2 , typename T3 >	
param2	BTWindow (T1 param1, T2 param2 , T3 param3) _BTWND_INITIALIZER_(param1) Object initialization (3 parameters).
template<typename T1 , typename T2 >	
	BTWindow (T1 param1, T2 param2) _BTWND_INITIALIZER_(param1) Object initialization (2 parameters).
template<typename T1 >	
	BTWindow (T1 param1) _BTWND_INITIALIZER_(param1) Object initialization (1 parameter).
	BTWindow (void) _BTWND_INITIALIZER_() Object initialization (0 parameters).
virtual WNDPROC	GetWindowProc (void) Overridden to return address of custom window procedure.

Protected Attributes

T5	param2
T4	param2
param2	param2
T5	param3
T4	param3
T5	param4
	T2
	T3
	T4

Private Member Functions

	LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers) Exception filter.
LRESULT	PrivWindowProc (HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam) This window procedure intercepts C++ exceptions.

Private Attributes

WNDPROC	m_pfnBaseWndProc
	Cached address of base window procedure.

Detailed Description

```
template<class BASE_CLASS>
class ATL::BTWindow< BASE_CLASS >
```

This class adds error handling to ATL/WTL windows. Only one form of exception handling is permitted per function, therefore this class uses two functions to catch C++ and Windows errors.

Constructor & Destructor Documentation

```
template<class BASE_CLASS>
ATL::BTWindow< BASE_CLASS >::BTWindow( void ) [protected]
```

Object initialization (0 parameters).

```
template<class BASE_CLASS>
template<typename T1>
ATL::BTWindow< BASE_CLASS >::BTWindow( T1 param1 ) [explicit]
```

Object initialization (1 parameter).

```
template<class BASE_CLASS>
template<typename T1, typename T2>
ATL::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2
) [protected]
```

Object initialization (2 parameters).

```
template<class BASE_CLASS>
template<typename T1, typename T2, typename T3>
param2 ATL::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2,
T3 param3
)
```

Object initialization (3 parameters).

```
template<class BASE_CLASS>
T4 ATL::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2,
T3 param3,
T4 param4
) [prot]
```

Type Constraints

```
template<class BASE_CLASS >
T5 ATL::BTWindow< BASE_CLASS >::BTWindow ( T1 param1,
                                T2 param2,
                                T3 param3,
                                T4 param4,
                                T5 param5
) [prot]
```

Type Constraints

Member Function Documentation

```
template<class BASE_CLASS >
```

```
WNDPROC ATL::BTWindow< BASE_CLASS >::GetWindowProc ( void )
```

Overridden to return address of custom window procedure.

Returns:

address of window procedure.

Type Constraints

```
template<class BASE_CLASS >
```

```
ATL::BTWindow< BASE_CLASS >::LONG ( CALLBACK * m_pfnFilter )
```

Exception filter.

```
template<class BASE_CLASS >
```

```
LRESULT ATL::BTWindow< BASE_CLASS >::PrivWindowProc ( HWND hWind
```

UINT nMsg,

WPARAM wParam,

LPARAM lParam)

This window procedure intercepts C++ exceptions.

Parameters:

hWnd - window handle.

uMsg - specifies the Windows message to be processed.

wParam - provides additional information used in processing the message.

lParam - provides additional information used in processing the message.

Returns:

the return value depends on the message.

References [BT_CppFilter\(\)](#), [BT_SetUserMessage\(\)](#),
[BT_SetUserMessageFromCode\(\)](#), and [ATL::BTWindow<](#)

BASE_CLASS >::m_pfnBaseWndProc.

Member Data Documentation

template<class BASE_CLASS >

WNDPROC ATL::BTWindow< BASE_CLASS >::m_pfnBaseWndProc

Cached address of base window procedure.

Referenced by **ATL::BTWindow< BASE_CLASS >::PrivWindowProc()**.

template<class BASE_CLASS >

T5 ATL::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

T4 ATL::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

param2 ATL::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

T5 ATL::BTWindow< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >

T4 ATL::BTWindow< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >

T5 ATL::BTWindow< BASE_CLASS >::param4 [protected]

template<class BASE_CLASS >

ATL::BTWindow< BASE_CLASS >::T2 [protected]

template<class BASE_CLASS >

ATL::BTWindow< BASE_CLASS >::T3 [protected]

template<class BASE_CLASS >

ATL::BTWindow< BASE_CLASS >::T4 [protected]

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Class List](#)[Class Index](#)[Class Members](#)

MFC::BTWindow

MFC::BTWindow< BASE_CLASS > Class Template Reference

[List of all members.](#)

Protected Member Functions

T5 **BTWindow** (T1 param1, **T2 param2**, **T3 param3**, **T4 param4**, T5 param5) **_BTWND_INITIALIZER_**(param1)

T4 **BTWindow** (T1 param1, **T2 param2**, **T3 param3**, **T4 param4**)
_BTWND_INITIALIZER_(param1)

T3 **BTWindow** (T1 param1, **T2 param2**, **T3 param3**)
_BTWND_INITIALIZER_(param1)

template<typename T1 , typename T2 >

BTWindow (T1 param1, **T2 param2**)

_BTWND_INITIALIZER_(param1)

Object initialization (2 parameters).

template<typename T1 >

BTWindow (T1 param1) **_BTWND_INITIALIZER_**(param1)

Object initialization (1 parameter).

BTWindow (void) **_BTWND_INITIALIZER_**()

Object initialization (0 parameters).

Protected Attributes

T5	param2
T4	param2
T3	param2
T5	param3
T4	param3
T5	param4
	T2
	T3
	T4

Private Member Functions

	LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers) Exception filter.
LRESULT	PrivWindowProc (UINT uMsg, WPARAM wParam, LPARAM lParam) This window procedure intercepts MFC exceptions.

Detailed Description

```
template<class BASE_CLASS>
class MFC::BTWindow< BASE_CLASS >
```

This class substitutes default **MFC** error handling. Only one form of exception handling is permitted per function, therefore this class uses two functions to catch C++ and Windows errors.

Constructor & Destructor Documentation

```
template<class BASE_CLASS>
MFC::BTWindow< BASE_CLASS >::BTWindow( void ) [protected]
```

Object initialization (0 parameters).

```
template<class BASE_CLASS>
template<typename T1>
MFC::BTWindow< BASE_CLASS >::BTWindow( T1 param1 ) [explicit]
```

Object initialization (1 parameter).

```
template<class BASE_CLASS>
template<typename T1, typename T2>
MFC::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2
) [protected]
```

Object initialization (2 parameters).

```
template<class BASE_CLASS>
T3 MFC::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2,
T3 param3
) [protected]
```

Type Constraints

```
template<class BASE_CLASS>
T4 MFC::BTWindow< BASE_CLASS >::BTWindow( T1 param1,
T2 param2,
T3 param3,
T4 param4
) [protected]
```

Type Constraints

```
template<class BASE_CLASS >
T5 MFC::BTWindow< BASE_CLASS >::BTWindow ( T1 param1,
                                T2 param2,
                                T3 param3,
                                T4 param4,
                                T5 param5
) [prot]
```

Type Constraints

Member Function Documentation

```
template<class BASE_CLASS >
MFC::BTWindow< BASE_CLASS >::LONG ( CALLBACK * m_pfnFilter )
```

Exception filter.

```
template<class BASE_CLASS >
LRESULT MFC::BTWindow< BASE_CLASS >::PrivWindowProc ( UIN
          WPA
          LPA
      )
```

This window procedure intercepts **MFC** exceptions.

Parameters:

- uMsg* - specifies the Windows message to be processed.
- wParam* - provides additional information used in processing the message.
- lParam* - provides additional information used in processing the message.

Returns:

the return value depends on the message.

Type Constraints

References **BT_CppFilter()**, and **BT_SetUserMessage()**.

Member Data Documentation

template<class BASE_CLASS >

T5 MFC::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

T4 MFC::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

T3 MFC::BTWindow< BASE_CLASS >::param2 [protected]

template<class BASE_CLASS >

T5 MFC::BTWindow< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >

T4 MFC::BTWindow< BASE_CLASS >::param3 [protected]

template<class BASE_CLASS >

T5 MFC::BTWindow< BASE_CLASS >::param4 [protected]

template<class BASE_CLASS >

MFC::BTWindow< BASE_CLASS >::T2 [protected]

template<class BASE_CLASS >

MFC::BTWindow< BASE_CLASS >::T3 [protected]

template<class BASE_CLASS >

MFC::BTWindow< BASE_CLASS >::T4 [protected]

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BUGTRAP_LOGFILEENTRY_tag Struct Reference

[Common type definitions]

Log file entry structure return by [BT_GetLogFileEntry\(\)](#) for regular log files. [More...](#)

List of all members.

Public Attributes

TCHAR	szLogFileName [MAX_PATH]
	Log file name.

Detailed Description

Log file entry structure return by [**BT_GetLogFileEntry\(\)**](#) for regular log files.

Member Data Documentation

TCHAR BUGTRAP_LOGFILEENTRY_tag::szLogFileName[MAX_PATH]

Log file name.

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BUGTRAP_REGEXPORTENTRY_tag Struct Reference

[Common type definitions]

Log file entry structure return by [BT_GetLogFileEntry\(\)](#) for log files exported from the registry. [More...](#)

[List of all members.](#)

Public Attributes

TCHAR	szLogFileName [MAX_PATH] Log file name.
TCHAR	szRegKey [MAX_PATH] Exported registry key.

Detailed Description

Log file entry structure return by [BT_GetLogFileEntry\(\)](#) for log files exported from the registry.

Member Data Documentation

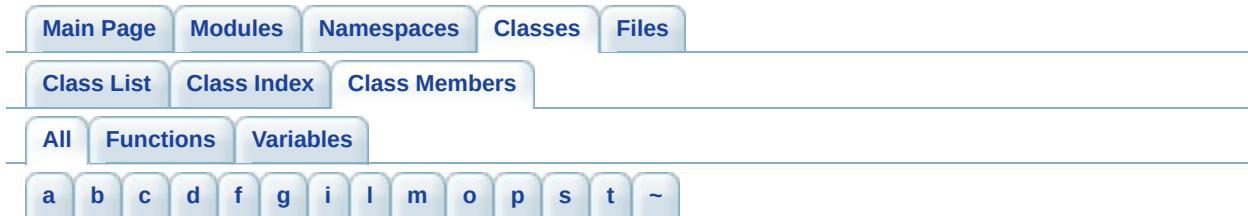
TCHAR BUGTRAP_REGEXPORTENTRY_tag::szLogFileName[MAX_PATH]

Log file name.

TCHAR BUGTRAP_REGEXPORTENTRY_tag::szRegKey[MAX_PATH]

Exported registry key.

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1



Here is a list of all class members with links to the classes they belong to:

- a -

- Append() : **BTTrace**
- AppendF() : **BTTrace**
- AppendV() : **BTTrace**
- Attach() : **BTTrace**

- b -

- BTDialog() : **ATL::BTDialog< BASE_CLASS >**
- BTTrace() : **BTTrace**
- BTWindow() : **ATL::BTWindow< BASE_CLASS >**,
MFC::BTWindow< BASE_CLASS >, **ATL::BTWindow< BASE_CLASS >**,
MFC::BTWindow< BASE_CLASS >, **ATL::BTWindow< BASE_CLASS >**

- c -

- Clear() : **BTTrace**
- Close() : **BTTrace**

- d -

- Detach() : **BTTrace**

- f -

- Flush() : **BTTrace**

- g -

- GetDefaultLogLevel() : **BTTrace**

- GetDialogProc() : **ATL::BTDialog< BASE_CLASS >**
- GetFileName() : **BTTrace**
- GetHandle() : **BTTrace**
- GetLogEchoMode() : **BTTrace**
- GetLogFlags() : **BTTrace**
- GetLogLevel() : **BTTrace**
- GetLogSizeInBytes() : **BTTrace**
- GetLogSizeInEntries() : **BTTrace**
- GetWindowProc() : **ATL::BTWindow< BASE_CLASS >**

- i -

- Insert() : **BTTrace**
- InsertF() : **BTTrace**
- InsertV() : **BTTrace**

- l -

- LONG() : **ATL::BTWindow< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >** , **ATL::BTDialog< BASE_CLASS >**

- m -

- m_eDefaultLogLevel : **BTTrace**
- m_iHandle : **BTTrace**
- m_pfnBaseDlgProc : **ATL::BTDialog< BASE_CLASS >**
- m_pfnBaseWndProc : **ATL::BTWindow< BASE_CLASS >**

- o -

- Open() : **BTTrace**
- operator=() : **BTTrace**

- p -

- param2 : **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDialog< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >** , **ATL::BTDialog< BASE_CLASS >** , **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDialog< BASE_CLASS >** ,

MFC::BTWindow< BASE_CLASS >

- param3 : **ATL::BTDIalog< BASE_CLASS >** , **ATL::BTWindow< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >** , **ATL::BTDIalog< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >**
- param4 : **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDIalog< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >**
- PrivDialogProc() : **ATL::BTDIalog< BASE_CLASS >**
- PrivWindowProc() : **ATL::BTWindow< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >**

- S -

- SetDefaultLogLevel() : **BTTrace**
- SetLogEchoMode() : **BTTrace**
- SetLogFlags() : **BTTrace**
- SetLogLevel() : **BTTrace**
- SetLogSizeInBytes() : **BTTrace**
- SetLogSizeInEntries() : **BTTrace**
- szLogFileName : **BUGTRAP_REGEXPORTENTRY_tag** , **BUGTRAP_LOGFILEENTRY_tag**
- szRegKey : **BUGTRAP_REGEXPORTENTRY_tag**

- t -

- T2 : **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDIalog< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >**
- T3 : **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDIalog< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >**
- T4 : **MFC::BTWindow< BASE_CLASS >** , **ATL::BTWindow< BASE_CLASS >** , **ATL::BTDIalog< BASE_CLASS >**

- ~ -

- ~BTTrace() : **BTTrace**

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[Namespace List](#)

Namespace List

Here is a list of all namespaces with brief descriptions:

ATL

MFC

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Namespace List](#)

ATL Namespace Reference

Classes

class	BTDialog
class	BTWindow

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Namespace List](#)

MFC Namespace Reference

Classes

class **BTWindow**

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[File List](#) [File Members](#)

File List

Here is a list of all files with brief descriptions:

BTAtlWindow.h	
BTMfcWindow.h	
BTTrace.h	
BugTrap.h	

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[File List](#) [File Members](#)

BTAtlWindow.h File Reference

Classes

class [ATL::BTDialog< BASE_CLASS >](#)

class [ATL::BTWindow< BASE_CLASS >](#)

Namespaces

namespace [ATL](#)

Defines

```
#define _BTDLG_INITIALIZER_ : m_pfnFilter(&BT_SehFilter),  
    m_pfnBaseDlgProc(BASE_CLASS::GetDialogProc()),  
    BASE_CLASS  
#define _BTWND_INITIALIZER_ : m_pfnFilter(&BT_SehFilter),  
    m_pfnBaseWndProc(BASE_CLASS::GetWindowProc()),  
    BASE_CLASS
```

Define Documentation

```
#define _BTDLG_INITIALIZER_ : m_pfnFilter(&BT_SehFilter), m_pf
```

```
#define _BTWND_INITIALIZER_ : m_pfnFilter(&BT_SehFilter), m_pf
```

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[File List](#) [File Members](#)

BTMfcWindow.h File Reference

Classes

class **MFC::BTWindow< BASE_CLASS >**

Namespaces

namespace [MFC](#)

Defines

```
#define _BTWND_INITIALIZER_ : m_pfnFilter(&BT_SehFilter),  
    BASE_CLASS
```

Define Documentation

```
#define _BTWND_INITIALIZER_ : m_pfnFilter(&BT_SehFilter), BASE
```

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  doxygen 1.5.7.1

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[File List](#) [File Members](#)

BTTrace.h File Reference

Classes

class **BTTrace**

C++ wrapper for tracing API. [More...](#)

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#) [Modules](#) [Namespaces](#) [Classes](#) [Files](#)

[File List](#) [File Members](#)

BugTrap.h File Reference

Classes

struct	BUGTRAP_LOGFILEENTRY_tag
	Log file entry structure return by BT_GetLogFileEntry() for regular log files. More...
struct	BUGTRAP_REGEXPORTENTRY_tag
	Log file entry structure return by BT_GetLogFileEntry() for log files exported from the registry. More...

Defines

#define	BT_DO_NOT_USE_DEFAULT_NAMESPACES
#define	BT_SetTerminate()
#define	BUGTRAP_API __declspec(dllexport)
#define	BUGTRAP_HTTP_PORT 80 Use this port for HTTP protocol.
#define	MiniDumpNoDump MAXDWORD Disables mini-dump generation in detailed mode.

Typedefs

typedef void(CALLBACK * BT_ErrHandler)(INT_P nErrorHandlerParam)	BT_ErrHandler Type definition of user-defined error handler that's called before and after main BugTrap calls.
typedef enum BUGTRAP_ACTIVITY_tag	BUGTRAP_ACTIVITY Type of action which is performed in response to an error.
typedef enum BUGTRAP_DIALOGMESSAGE_tag	BUGTRAP_DIALOGMESSAGE Type of user defined message displayed on the screen.
typedef enum BUGTRAP_EXITMODE_tag	BUGTRAP_EXITMODE These flags control application termination mode: what the application needs to be terminated after error handling and what control should passed to exception handler.
typedef enum BUGTRAP_FLAGS_tag	BUGTRAP_FLAGS Different BugTrap options: you can use any combination of these flags.
typedef enum BUGTRAP_LOGECHOTYPE_tag	BUGTRAP_LOGECHOTYPE Type of log echo mode.
typedef struct BUGTRAP_LOGFILEENTRY_tag	BUGTRAP_LOGFILEENTRY Log file entry structure returned by BT_GetLogFileEntry() function.

	regular log files.
typedef enum BUGTRAP_LOGFLAGS_tag	BUGTRAP_LOGFLAGS Set of log file options.
typedef enum BUGTRAP_LOGFORMAT_tag	BUGTRAP_LOGFORMAT Format of log file.
typedef enum BUGTRAP_LOGLEVEL_tag	BUGTRAP_LOGLEVEL Set of available log levels
typedef enum BUGTRAP_LOGTYPE_tag	BUGTRAP_LOGTYPE Type of log file attached to report.
typedef struct BUGTRAP_REGEXPORTENTRY_tag	BUGTRAP_REGEXPORTENTRY Log file entry structure returned by BT_GetLogFileEntry() function for files exported from the report.
typedef enum BUGTRAP_REPORTFORMAT_tag	BUGTRAP_REPORTFORMAT Format of error report.

Enumerations

enum	BUGTRAP_ACTIVITY_tag { BTA_SHOWUI = 1, BTA_SAVEREPORT = 2, BTA_MAILREPORT = 3, BTA_SENDREPORT = 4 }
	Type of action which is performed in response to the error. More...
enum	BUGTRAP_DIALOGMESSAGE_tag { BTDM_INTRO1 = 1, BTDM_INTRO2 = 2 }
	Type of user defined message displayed on the screen. More...
enum	BUGTRAP_EXITMODE_tag { BTEM_TERMINATEAPP , BTEM_CONTINUESEARCH , BTEM_EXECUTEHANDLER }
	These flags control application termination mode: whatever the application needs to be terminated after error handling, control should passed to another exception handler. More...
enum	BUGTRAP_FLAGS_tag { BTF_NONE = 0x00, BTF_DETAILEDMODE = 0x01, BTF_EDITMAIL = 0x02, BTF_ATTACHREPORT = 0x04, BTF_LISTPROCESSES = 0x08, BTF_SHOWADVANCEDUI = 0x10, BTF_SCREENCAPTURE = 0x20, BTF_INTERCEPTSUEF = 0x80 }
	Different BugTrap options. You can use any combinations of these flags. More...
enum	BUGTRAP_LOGECHOTYPE_tag { BTLE_NONE = 0x00, BTLE_STDOUT = 0x01, BTLE_STDERR = 0x02, BTLE_DBGOUT = 0x04 }
	Type of log echo mode. More...
enum	BUGTRAP_LOGFLAGS_tag { BTLF_NONE = 0x00, BTLF_SHOWLOGLEVEL = 0x01, BTLF_SHOWTIMESTAMP = 0x02 }
	Set of log file options. More...
enum	BUGTRAP_LOGFORMAT_tag { BTLF_XML = 1, BTLF_TEXT = 2, BTLF_STREAM = 3 }

	Format of log file. More...
enum	BUGTRAP_LOGLEVEL_tag { BTLL_NONE = 0, BTLL_ERROR = 1, BTLL_WARNING = 2, BTLL_IMPORTANT = 3, BTLL_INFO = 4, BTLL_VERBOSE = 5, BTLL_ALL = BTLL_INFO }
	Set of available log levels. More...
enum	BUGTRAP_LOGTYPE_tag { BTLT_LOGFILE , BTLT_REGEXPORT }
	Type of log file attached to the report. More...
enum	BUGTRAP_REPORTFORMAT_tag { BTRF_XML = 1, BTRF_TEXT = 2 }
	Format of error report. More...

Functions

BUGTRAP_API void APIENTRY	BT_AddLogFile (LPCTSTR pszLogFile)
	Add custom log entry to the of custom log files attached bug report.
BUGTRAP_API void APIENTRY	BT_AddRegFile (LPCTSTR pszRegFile, LPCTSTR pszRegKey)
	Add custom log entry to the of custom log files attached bug report.
BUGTRAP_API BOOL APIENTRY	BT_AppLogEntry (INT_P iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR pszAppend entry to the end of custom log file. This function is thread safe.
BUGTRAP_API BOOL CDECL	BT_AppLogEntryF (INT_P iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR pszFormat,...)
	Append entry to the end of custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_AppLogEntryV (INT_P iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR pszFormat, va_list argList)
	Append entry to the end of custom log file. This function is thread safe.
BUGTRAP_API void CDECL	BT_CallCppFilter (void)

		Simulates access violation C++ application. Used by set_terminate().
BUGTRAP_API void CDECL	BT_CallSehFilter (void)	Simulates access violation C++ application. Used by set_terminate().
BUGTRAP_API BOOL APIENTRY	BT_ClearLog (INT_PTR iHandle)	Clear log file. This function is thread safe.
BUGTRAP_API void APIENTRY	BT_ClearLogFile (void)	Clear the list of custom log files attached to bug report.
BUGTRAP_API BOOL APIENTRY	BT_CloseLogFile (INT_PTR iHandle)	Close custom log file. This function is thread safe.
BUGTRAP_API LONG CALLBACK	BT_CppFilter (PEXCEPTION_POINTER pExceptionPointers)	Executes C++ exception filters.
BUGTRAP_API void APIENTRY	BT_DeleteLogFile (LPCTSTR pszLogFile)	Delete custom log entry from list of custom log files attached to bug report.
BUGTRAP_API int APIENTRY	BT_ExportRegistryKey (LPCTSTR pszRegFile, LPCTSTR pszRegKey)	Export registry key to the file.
BUGTRAP_API BOOL APIENTRY	BT_FlushLogFile (INT_PTR iHandle)	Flush contents of the log file.

BUGTRAP_API BUGTRAP_ACTIVITY APIENTRY	BT_GetActivityType (void) Get the type of action which performed in response to the error.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetAppName (void) Get application name.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetAppVersion (void) Get application version number.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetDialogMessage (BUGTRAP_DIALOGMESSAGE *eDialogMessage) Get user defined message displayed on the screen.
BUGTRAP_API DWORD APIENTRY	BT_GetDumpType (void) Get the type of produced report dump in detailed mode.
BUGTRAP_API BUGTRAP_EXITMODE APIENTRY	BT_GetExitMode (void) Get application termination mode.
BUGTRAP_API DWORD APIENTRY	BT_GetFlags (void) Get current BugTrap options.
BUGTRAP_API DWORD APIENTRY	BT_GetLogEchoMode (INT_PTR iHandle) Get echo mode.
BUGTRAP_API DWORD APIENTRY	BT_GetLogFileEntry (INT nLogFileIndexOrName, BOOL bGetByIndex, BUGTRAP_LOGTYPE *peLogType, PDWORD pdwLogEntrySize, PVOID pLogEntry)

		This function enumerates log entries that will be attached to the report.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetLogFileName (INT_PTR iHandle)	Get custom log file name. This function is thread safe.
BUGTRAP_API DWORD APIENTRY	BT_GetLogFilesCount (void)	Get number of log files attached to the report.
BUGTRAP_API DWORD APIENTRY	BT_GetLogFlags (INT_PTR iHandle)	Return true if time stamp is added to every log entry.
BUGTRAP_API BUGTRAP_LOGLEVEL APIENTRY	BT_GetLogLevel (INT_PTR iHandle)	Return minimal log level accepted by tracing functions.
BUGTRAP_API DWORD APIENTRY	BT_GetLogSizeInBytes (INT_PTR iHandle)	Get maximum log file size in bytes. This function is thread safe.
BUGTRAP_API DWORD APIENTRY	BT_GetLogSizeInEntries (INT_PTR iHandle)	Get maximum log file size in records. This function is thread safe.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetMailProfile (void)	Get the name of MAPI profile.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetNotificationEMail	Get error notification e-mail address. BugTrap Server can automatically notify products.

		support by e-mail about new reports.
BUGTRAP_API BT_ErrHandler APIENTRY	BT_GetPostErrorHandler (void)	Get address of error handler called after BugTrap dialog.
BUGTRAP_API BT_ErrHandler APIENTRY	BT_GetPreErrorHandler (void)	Get address of error handler called before BugTrap dialog.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetReportFilePath (void)	Get the path of error report.
BUGTRAP_API BUGTRAP_REPORTFORMAT APIENTRY	BT_GetReportFormat (void)	Get format of error report.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportEmail (void)	Get product support e-mail address.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportHost (void)	Get host name of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API SHORT APIENTRY	BT_GetSupportPort (void)	Get port number of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API LPCTSTR APIENTRY	BT_GetSupportURL (void)	Get HTTP address of product support site.
BUGTRAP_API LPCTSTR APIENTRY	BT GetUserMessage (void)	Get user defined message. message may be printed to file.

BUGTRAP_API BOOL APIENTRY	BT_InsLogEntry (INT_PT iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR psz Insert entry into the beginn custom log file. This functio thread safe.
BUGTRAP_API BOOL CDECL	BT_InsLogEntryF (INT_F iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR pszFormat,...) Insert entry into the beginn custom log file. This functio thread safe.
BUGTRAP_API BOOL APIENTRY	BT_InsLogEntryV (INT_F iHandle, BUGTRAP_LOG eLogLevel, LPCTSTR pszFormat, va_list argList) Insert entry into the beginn custom log file. This functio thread safe.
BUGTRAP_API LPTOP_LEVEL_EXCEPTION_FILTER APIENTRY	BT_InstallSehFilter (void)
BUGTRAP_API void	BT_InterceptSUEF (HMO hModule, BOOL bOverride Explicitly intercepts SetUnhandledExceptionFi in a module. This function be useful for dynamically lo modules.
BUGTRAP_API BOOL APIENTRY	BT_MailSnapshot (void) Take a snapshot of progra memory and e-mail it.
BUGTRAP_API BOOL APIENTRY	BT_MailSnapshotEx (PEXCEPTION_POINTER

		pExceptionPointers)
		Take a snapshot of program memory and e-mail it.
BUGTRAP_API INT_PTR APIENTRY	BT_OpenLogFile (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)	Open custom log file. This function is thread safe.
BUGTRAP_API BOOL APIENTRY	BT_ReadVersionInfo (HMODULE hModule)	Read application name and version number from version info block.
BUGTRAP_API BOOL APIENTRY	BT_SaveSnapshot (PCTSTR pszFileName)	Take a snapshot of program memory and save it to a file.
BUGTRAP_API BOOL APIENTRY	BT_SaveSnapshotEx (PEXCEPTION_POINTER pExceptionPointers, PCTSTR pszFileName)	Take a snapshot of program memory and save it to a file.
BUGTRAP_API LONG CALLBACK	BT_SehFilter (PEXCEPTION_POINTER pExceptionPointers)	Executes structured exception filter.
BUGTRAP_API BOOL APIENTRY	BT_SendSnapshot (void)	Take a snapshot of program memory and send over network.
BUGTRAP_API BOOL APIENTRY	BT_SendSnapshotEx (PEXCEPTION_POINTER	

		pExceptionPointers) Take a snapshot of program memory and send over network.
BUGTRAP_API void APIENTRY	BT_SetActivityType (BUGTRAP_ACTIVITY eActivityType)	Set the type of action which performed in response to the error.
BUGTRAP_API void APIENTRY	BT_SetAppName (LPCTSTR pszAppName)	Set application name of the project where BugTrap is used.
BUGTRAP_API void APIENTRY	BT_SetAppVersion (LPCSTR pszAppVersion)	Set application version number.
BUGTRAP_API void APIENTRY	BT_SetDialogMessage (BUGTRAP_DIALOGMESSAGE eDialogMessage, LPCTSTR pszMessage)	Set user defined message displayed on the screen.
BUGTRAP_API void APIENTRY	BT_SetDumpType (DWORD dwDumpType)	Set the type of produced memory dump in detailed mode.
BUGTRAP_API void APIENTRY	BT_SetExitMode (BUGTRAP_EXITMODE eExitMode)	Set application termination mode.
BUGTRAP_API void APIENTRY	BT_SetFlags (DWORD dwFlags)	Set various BugTrap options.
BUGTRAP_API BOOL APIENTRY	BT_SetLogEchoMode	

		(INT_PTR iHandle, DWORD dwLogEchoMode) Set echo mode.
BUGTRAP_API BOOL APIENTRY	BT_SetLogFlags (INT_PTR iHandle, DWORD dwLogFileFlags) Set true if time stamp is added to every log entry.	
BUGTRAP_API BOOL APIENTRY	BT_SetLogLevel (INT_PTR iHandle, BUGTRAP_LOG eLogLevel) Set minimal log level accepted by tracing functions.	
BUGTRAP_API BOOL APIENTRY	BT_SetLogSizeInBytes (INT_PTR iHandle, DWORD dwLogSizeInBytes) Set maximum log file size in bytes. This function is thread safe.	
BUGTRAP_API BOOL APIENTRY	BT_SetLogSizeInEntries (INT_PTR iHandle, DWORD dwLogSizeInEntries) Set maximum log file size in records. This function is thread safe.	
BUGTRAP_API void APIENTRY	BT_SetMailProfile (LPCTSTR pszMailProfile, LPCTSTR pszMailPassword) Set the name and password of the MAPI profile.	
BUGTRAP_API void APIENTRY	BT_SetNotificationEMail (LPCTSTR pszNotificationEMail) Set error notification e-mail address. BugTrap Server will automatically notify products.	

		support by e-mail about new reports.
BUGTRAP_API void APIENTRY	BT_SetPostErrorHandler (BT_ErrHandler pfnPostErrorHandler, INT_PTR nPostErrorHandlerParam)	Set address of error handler called after BugTrap dialog.
BUGTRAP_API void APIENTRY	BT_SetPreErrorHandler (BT_ErrHandler pfnPreErrorHandler, INT_PTR nPreErrorHandlerParam)	Set address of error handler called before BugTrap dialog.
BUGTRAP_API void APIENTRY	BT_SetReportFilePath (LPCTSTR pszReportFilePath)	Set the path of error report function has effect only for <i>BTA_SAVEREPORT</i> activation.
BUGTRAP_API void APIENTRY	BT_SetReportFormat (BUGTRAP_REPORTFORMAT eReportFormat)	Set format of error report.
BUGTRAP_API void APIENTRY	BT_SetSupportEMail (LPCTSTR pszSupportEMail)	Set product support e-mail address.
BUGTRAP_API void APIENTRY	BT_SetSupportHost (LPCTSTR pszSupportHost)	Set host name (address) of BugTrap Server. This server will automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetSupportPort (SHORT nSupportPort)	

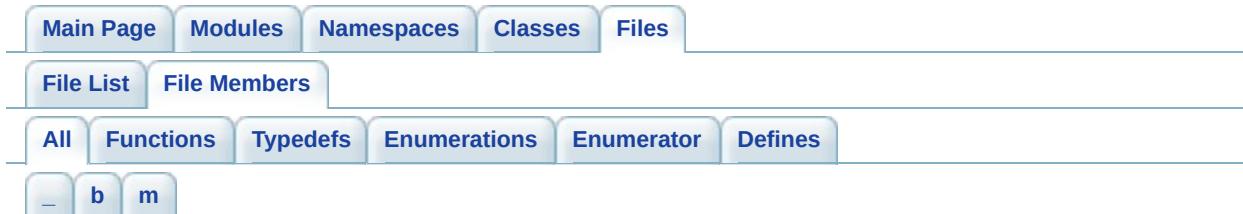
	nSupportPort) Set port number of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetSupportServer (LPCTSTR pszSupportHost SHORT nSupportPort) Set host name (address) and port number of BugTrap Server. This server can automatically gather bug reports for the application.
BUGTRAP_API void APIENTRY	BT_SetSupportURL (LPCSTR pszSupportURL) Set HTTP address of production support site.
BUGTRAP_API void APIENTRY	BT_SetUserMessage (LPCTSTR pszUserMessage) Set user defined message. The message may be printed to a file.
BUGTRAP_API void APIENTRY	BT_SetUserMessageFromError (DWORD dwErrorCode) Set user defined message from error code. The message may be printed to a file.
BUGTRAP_API void APIENTRY	BT_UninstallSehFilter (void)

Define Documentation

```
#define BT_DO_NOT_USE_DEFAULT_NAMESPACES
```

```
#define BUGTRAP_API __declspec(dllimport)
```

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1



Here is a list of all file members with links to the files they belong to:

- _ -

- `_BTDLG_INITIALIZER_` : [BTAtlWindow.h](#)
- `_BTWND_INITIALIZER_` : [BTMfcWindow.h](#) , [BTAtlWindow.h](#)

- b -

- `BT_AddLogFile()` : [BugTrap.h](#)
- `BT_AddRegFile()` : [BugTrap.h](#)
- `BT_AppLogEntry()` : [BugTrap.h](#)
- `BT_AppLogEntryF()` : [BugTrap.h](#)
- `BT_AppLogEntryV()` : [BugTrap.h](#)
- `BT_CallCppFilter()` : [BugTrap.h](#)
- `BT_CallSehFilter()` : [BugTrap.h](#)
- `BT_ClearLog()` : [BugTrap.h](#)
- `BT_ClearLogFile()` : [BugTrap.h](#)
- `BT_CppFilter()` : [BugTrap.h](#)
- `BT_DeleteLogFile()` : [BugTrap.h](#)
- `BT_DO_NOT_USE_DEFAULT_NAMESPACES` : [BugTrap.h](#)
- `BT_Handler` : [BugTrap.h](#)
- `BT_ExportRegistryKey()` : [BugTrap.h](#)
- `BT_FlushLogFile()` : [BugTrap.h](#)
- `BT_GetActivityType()` : [BugTrap.h](#)
- `BT_GetAppName()` : [BugTrap.h](#)
- `BT_GetAppVersion()` : [BugTrap.h](#)
- `BT_GetDialogMessage()` : [BugTrap.h](#)
- `BT_GetDumpType()` : [BugTrap.h](#)
- `BT_GetExitMode()` : [BugTrap.h](#)
- `BT_GetFlags()` : [BugTrap.h](#)
- `BT_GetLogEchoMode()` : [BugTrap.h](#)

- BT_GetLogFileEntry() : **BugTrap.h**
- BT_GetLogFileName() : **BugTrap.h**
- BT_GetLogFilesCount() : **BugTrap.h**
- BT_GetLogFlags() : **BugTrap.h**
- BT_GetLogLevel() : **BugTrap.h**
- BT_GetLogSizeInBytes() : **BugTrap.h**
- BT_GetLogSizeInEntries() : **BugTrap.h**
- BT_GetMailProfile() : **BugTrap.h**
- BT_GetNotificationEMail() : **BugTrap.h**
- BT_GetPostErrHandler() : **BugTrap.h**
- BT_GetPreErrHandler() : **BugTrap.h**
- BT_GetReportFilePath() : **BugTrap.h**
- BT_GetReportFormat() : **BugTrap.h**
- BT_GetSupportEMail() : **BugTrap.h**
- BT_GetSupportHost() : **BugTrap.h**
- BT_GetSupportPort() : **BugTrap.h**
- BT_GetSupportURL() : **BugTrap.h**
- BT GetUserMessage() : **BugTrap.h**
- BT_InsLogEntry() : **BugTrap.h**
- BT_InsLogEntryF() : **BugTrap.h**
- BT_InsLogEntryV() : **BugTrap.h**
- BT_InstallSehFilter() : **BugTrap.h**
- BT_InterceptSUEF() : **BugTrap.h**
- BT_MailSnapshot() : **BugTrap.h**
- BT_MailSnapshotEx() : **BugTrap.h**
- BT_OpenLogFile() : **BugTrap.h**
- BT_ReadVersionInfo() : **BugTrap.h**
- BT_SaveSnapshot() : **BugTrap.h**
- BT_SaveSnapshotEx() : **BugTrap.h**
- BT_SehFilter() : **BugTrap.h**
- BT_SendSnapshot() : **BugTrap.h**
- BT_SendSnapshotEx() : **BugTrap.h**
- BT_SetActivityType() : **BugTrap.h**
- BT_SetAppName() : **BugTrap.h**
- BT_SetAppVersion() : **BugTrap.h**
- BT_SetDialogMessage() : **BugTrap.h**
- BT_SetDumpType() : **BugTrap.h**

- BT_SetExitMode() : **BugTrap.h**
- BT_SetFlags() : **BugTrap.h**
- BT_SetLogEchoMode() : **BugTrap.h**
- BT_SetLogFlags() : **BugTrap.h**
- BT_SetLogLevel() : **BugTrap.h**
- BT_SetLogSizeInBytes() : **BugTrap.h**
- BT_SetLogSizeInEntries() : **BugTrap.h**
- BT_SetMailProfile() : **BugTrap.h**
- BT_SetNotificationEMail() : **BugTrap.h**
- BT_SetPostErrHandler() : **BugTrap.h**
- BT_SetPreErrHandler() : **BugTrap.h**
- BT_SetReportFilePath() : **BugTrap.h**
- BT_SetReportFormat() : **BugTrap.h**
- BT_SetSupportEMail() : **BugTrap.h**
- BT_SetSupportHost() : **BugTrap.h**
- BT_SetSupportPort() : **BugTrap.h**
- BT_SetSupportServer() : **BugTrap.h**
- BT_SetSupportURL() : **BugTrap.h**
- BT_SetTerminate : **BugTrap.h**
- BT_SetUserMessage() : **BugTrap.h**
- BT_SetUserMessageFromCode() : **BugTrap.h**
- BT_UninstallSehFilter() : **BugTrap.h**
- BTA_MAILREPORT : **BugTrap.h**
- BTA_SAVEREPORT : **BugTrap.h**
- BTA_SENDREPORT : **BugTrap.h**
- BTA_SHOWUI : **BugTrap.h**
- BTDM_INTRO1 : **BugTrap.h**
- BTDM_INTRO2 : **BugTrap.h**
- BTEM_CONTINUESEARCH : **BugTrap.h**
- BTEM_EXECUTEHANDLER : **BugTrap.h**
- BTEM_TERMINATEAPP : **BugTrap.h**
- BTF_ATTACHREPORT : **BugTrap.h**
- BTF_DETAILEDMODE : **BugTrap.h**
- BTF_EDITMAIL : **BugTrap.h**
- BTF_INTERCEPTSUEF : **BugTrap.h**
- BTF_LISTPROCESSES : **BugTrap.h**
- BTF_NONE : **BugTrap.h**

- BTF_SCREENCAPTURE : **BugTrap.h**
- BTF_SHOWADVANCEDUI : **BugTrap.h**
- BTLE_DBGOUT : **BugTrap.h**
- BTLE_NONE : **BugTrap.h**
- BTLE_STDERR : **BugTrap.h**
- BTLE_STDOUT : **BugTrap.h**
- BTLF_NONE : **BugTrap.h**
- BTLF_SHOWLOGLEVEL : **BugTrap.h**
- BTLF_SHOWTIMESTAMP : **BugTrap.h**
- BTLF_STREAM : **BugTrap.h**
- BTLF_TEXT : **BugTrap.h**
- BTLF_XML : **BugTrap.h**
- BTLL_ALL : **BugTrap.h**
- BTLL_ERROR : **BugTrap.h**
- BTLL_IMPORTANT : **BugTrap.h**
- BTLL_INFO : **BugTrap.h**
- BTLL_NONE : **BugTrap.h**
- BTLL_VERBOSE : **BugTrap.h**
- BTLL_WARNING : **BugTrap.h**
- BTLT_LOGFILE : **BugTrap.h**
- BTLT_REGEXPORT : **BugTrap.h**
- BTRF_TEXT : **BugTrap.h**
- BTRF_XML : **BugTrap.h**
- BUGTRAP_ACTIVITY : **BugTrap.h**
- BUGTRAP_ACTIVITY_tag : **BugTrap.h**
- BUGTRAP_API : **BugTrap.h**
- BUGTRAP_DIALOGMESSAGE : **BugTrap.h**
- BUGTRAP_DIALOGMESSAGE_tag : **BugTrap.h**
- BUGTRAP_EXITMODE : **BugTrap.h**
- BUGTRAP_EXITMODE_tag : **BugTrap.h**
- BUGTRAP_FLAGS : **BugTrap.h**
- BUGTRAP_FLAGS_tag : **BugTrap.h**
- BUGTRAP_HTTP_PORT : **BugTrap.h**
- BUGTRAP_LOGECHOTYPE : **BugTrap.h**
- BUGTRAP_LOGECHOTYPE_tag : **BugTrap.h**
- BUGTRAP_LOGFILEENTRY : **BugTrap.h**
- BUGTRAP_LOGFLAGS : **BugTrap.h**

- BUGTRAP_LOGFLAGS_tag : **BugTrap.h**
- BUGTRAP_LOGFORMAT : **BugTrap.h**
- BUGTRAP_LOGFORMAT_tag : **BugTrap.h**
- BUGTRAP_LOGLEVEL : **BugTrap.h**
- BUGTRAP_LOGLEVEL_tag : **BugTrap.h**
- BUGTRAP_LOGTYPE : **BugTrap.h**
- BUGTRAP_LOGTYPE_tag : **BugTrap.h**
- BUGTRAP_REGEXPORTENTRY : **BugTrap.h**
- BUGTRAP_REPORTFORMAT : **BugTrap.h**
- BUGTRAP_REPORTFORMAT_tag : **BugTrap.h**

- m -

- MiniDumpNoDump : **BugTrap.h**

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)

BugTrap Documentation

1.0

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[Class List](#)[Class Index](#)[Class Members](#)

Class Index

B

B

[BTDialog \(ATL\)](#)

[BTTrace](#)

[BTWindow \(MFC\)](#)

[BTWindow \(ATL\)](#)

[BUGTRAP_LOGFILEENTRY_tag](#)

[BUGTRAP_REGEXPORTENTRY_tag](#)

B

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

ATL::BTDialog< BASE_CLASS > Member List

This is the complete list of members for **ATL::BTDialog< BASE_CLASS >**, including all inherited members.

BTDialog (void) _BTDLG_INITIALIZER_()	ATL::BTDialog< BASE_CLASS >	[protected]
BTDialog (T1 param1) _BTDLG_INITIALIZER_(param1)	ATL::BTDialog< BASE_CLASS >	[explicit, protected]
BTDialog (T1 param1, T2 param2) _BTDLG_INITIALIZER_(param1	ATL::BTDialog< BASE_CLASS >	[protected]
BTDialog (T1 param1, T2 param2, T3 param3) _BTDLG_INITIALIZER_(param1	ATL::BTDialog< BASE_CLASS >	[protected]
BTDialog (T1 param1, T2 param2, T3 param3, T4 param4) _BTDLG_INITIALIZER_(param1	ATL::BTDialog< BASE_CLASS >	[protected]
BTDialog (T1 param1, T2 param2, T3 param3, T4 param4, T5 param5) _BTDLG_INITIALIZER_(param1	ATL::BTDialog< BASE_CLASS >	[protected]
GetDialogProc (void)	ATL::BTDialog< BASE_CLASS >	[protected, virtual]
LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers)	ATL::BTDialog< BASE_CLASS >	[private]
m_pfnBaseDlgProc	ATL::BTDialog< BASE_CLASS >	[private]
param2	ATL::BTDialog< BASE_CLASS >	[protected]

param2	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
param2	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
param3	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
param3	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
param4	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
PrivDialogProc (HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)	<code>ATL::BTDialog< BASE_CLASS ></code>	[private]
T2	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
T3	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]
T4	<code>ATL::BTDialog< BASE_CLASS ></code>	[protected]

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BTTrace Member List

This is the complete list of members for **BTTrace**, including all inherited members.

Append(BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry) const	BTTrace	
Append(LPCTSTR pszEntry) const	BTTrace	
AppendF(BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...) const	BTTrace	
AppendF(LPCTSTR pszFormat,...) const	BTTrace	
AppendV(BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList) const	BTTrace	
AppendV(LPCTSTR pszFormat, va_list argList) const	BTTrace	
Attach(INT_PTR iHandle)	BTTrace	
BTTrace(void)	BTTrace	
BTTrace(INT_PTR iHandle)	BTTrace	
BTTrace(LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)	BTTrace	
BTTrace(const BTTrace &rTrace)	BTTrace	[private]
Clear(void) const	BTTrace	
Close(void)	BTTrace	
Detach(void)	BTTrace	
Flush(void) const	BTTrace	
GetDefaultLogLevel(void) const	BTTrace	
GetFileName(void) const	BTTrace	
GetHandle(void) const	BTTrace	
GetLogEchoMode(void) const	BTTrace	
GetLogFlags(void) const	BTTrace	
GetLogLevel(void) const	BTTrace	
GetLogSizeInBytes(void) const	BTTrace	

GetLogSizeInEntries (void) const	BTTrace	
Insert (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszEntry) const	BTTrace	
Insert (LPCTSTR pszEntry) const	BTTrace	
InsertF (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat,...) const	BTTrace	
InsertF (LPCTSTR pszFormat,...) const	BTTrace	
InsertV (BUGTRAP_LOGLEVEL eLogLevel, LPCTSTR pszFormat, va_list argList) const	BTTrace	
InsertV (LPCTSTR pszFormat, va_list argList) const	BTTrace	
m_eDefaultLogLevel	BTTrace	[private]
m_iHandle	BTTrace	[private]
Open (LPCTSTR pszLogFileName, BUGTRAP_LOGFORMAT eLogFormat)	BTTrace	
operator= (const BTTrace &rTrace)	BTTrace	[private]
SetDefaultLogLevel (BUGTRAP_LOGLEVEL eDefaultLogLevel)	BTTrace	
SetLogEchoMode (DWORD dwLogEchoMode) const	BTTrace	
SetLogFlags (DWORD dwLogFlags) const	BTTrace	
SetLogLevel (BUGTRAP_LOGLEVEL eLogLevel) const	BTTrace	
SetLogSizeInBytes (DWORD dwSize) const	BTTrace	
SetLogSizeInEntries (DWORD dwSize) const	BTTrace	
~BTTrace (void)	BTTrace	

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

ATL::BTWindow< BASE_CLASS > Member List

This is the complete list of members for **ATL::BTWindow< BASE_CLASS >**, including all inherited members.

BTWindow (void) <u>_BTWND_INITIALIZER_()</u>	ATL::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1) <u>_BTWND_INITIALIZER_(param1)</u>	ATL::BTWindow< BASE_CLASS >	[explicit, protected]
BTWindow (T1 param1, T2 param2) <u>_BTWND_INITIALIZER_(param1)</u>	ATL::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3) <u>_BTWND_INITIALIZER_(param1)</u>	ATL::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3, T4 param4) <u>_BTWND_INITIALIZER_(param1)</u>	ATL::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3, T4 param4, T5 param5) <u>_BTWND_INITIALIZER_(param1)</u>	ATL::BTWindow< BASE_CLASS >	[protected]
GetWindowProc (void)	ATL::BTWindow< BASE_CLASS >	[protected, virtual]
LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers)	ATL::BTWindow< BASE_CLASS >	[private]
m_pfnBaseWndProc	ATL::BTWindow< BASE_CLASS >	[private]
param2	ATL::BTWindow< BASE_CLASS >	[protected]
param2	ATL::BTWindow< BASE_CLASS >	[protected]
param2	ATL::BTWindow< BASE_CLASS >	[protected]
	ATL::BTWindow<	

param3	BASE_CLASS >	[protected]
param3	ATL::BTWindow< BASE_CLASS >	[protected]
param4	ATL::BTWindow< BASE_CLASS >	[protected]
PrivWindowProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)	ATL::BTWindow< BASE_CLASS >	[private]
T2	ATL::BTWindow< BASE_CLASS >	[protected]
T3	ATL::BTWindow< BASE_CLASS >	[protected]
T4	ATL::BTWindow< BASE_CLASS >	[protected]

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

MFC::BTWindow< BASE_CLASS > Member List

This is the complete list of members for **MFC::BTWindow< BASE_CLASS >**, including all inherited members.

BTWindow (void) _BTWND_INITIALIZER_()	MFC::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1) _BTWND_INITIALIZER_(param1)	MFC::BTWindow< BASE_CLASS >	[explicit, protected]
BTWindow (T1 param1, T2 param2) _BTWND_INITIALIZER_(param1)	MFC::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3) _BTWND_INITIALIZER_(param1	MFC::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3, T4 param4) _BTWND_INITIALIZER_(param1	MFC::BTWindow< BASE_CLASS >	[protected]
BTWindow (T1 param1, T2 param2, T3 param3, T4 param4, T5 param5) _BTWND_INITIALIZER_(param1	MFC::BTWindow< BASE_CLASS >	[protected]
LONG (CALLBACK *m_pfnFilter) (PEXCEPTION_POINTERS pExceptionPointers)	MFC::BTWindow< BASE_CLASS >	[private]
param2	MFC::BTWindow< BASE_CLASS >	[protected]
param2	MFC::BTWindow< BASE_CLASS >	[protected]
param2	MFC::BTWindow< BASE_CLASS >	[protected]
param3	MFC::BTWindow< BASE_CLASS >	[protected]
param3	MFC::BTWindow< BASE_CLASS >	[protected]
	MFC::BTWindow<	

param4	BASE_CLASS >	[protected]
PrivWindowProc (UINT uMsg, WPARAM wParam, LPARAM lParam)	MFC::BTWindow< BASE_CLASS >	[private]
T2	MFC::BTWindow< BASE_CLASS >	[protected]
T3	MFC::BTWindow< BASE_CLASS >	[protected]
T4	MFC::BTWindow< BASE_CLASS >	[protected]

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BUGTRAP_LOGFILEENTRY_tag Member List

This is the complete list of members for
BUGTRAP_LOGFILEENTRY_tag, including all inherited members.

szLogFileName **BUGTRAP_LOGFILEENTRY_tag**

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

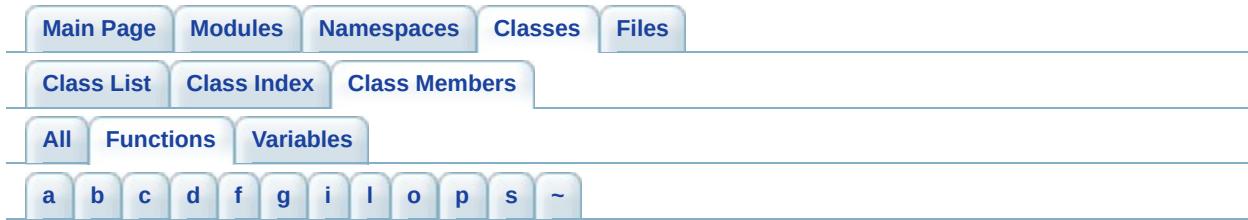
Main Page	Modules	Namespaces	Classes	Files
Class List	Class Index	Class Members		

BUGTRAP_REGEXPORTENTRY_tag Member List

This is the complete list of members for **BUGTRAP_REGEXPORTENTRY_tag**, including all inherited members.

szLogFileName	BUGTRAP_REGEXPORTENTRY_tag
szRegKey	BUGTRAP_REGEXPORTENTRY_tag

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1



- a -

- Append() : **BTTrace**
- AppendF() : **BTTrace**
- AppendV() : **BTTrace**
- Attach() : **BTTrace**

- b -

- BTDialog() : **ATL::BTDialog< BASE_CLASS >**
- BTTrace() : **BTTrace**
- BTWindow() : **ATL::BTWindow< BASE_CLASS >**,
MFC::BTWindow< BASE_CLASS >, **ATL::BTWindow< BASE_CLASS >**,
MFC::BTWindow< BASE_CLASS >, **ATL::BTWindow< BASE_CLASS >**

- c -

- Clear() : **BTTrace**
- Close() : **BTTrace**

- d -

- Detach() : **BTTrace**

- f -

- Flush() : **BTTrace**

- g -

- GetDefaultLogLevel() : **BTTrace**

- GetDialogProc() : **ATL::BTDialog< BASE_CLASS >**
- GetFileName() : **BTTrace**
- GetHandle() : **BTTrace**
- GetLogEchoMode() : **BTTrace**
- GetLogFlags() : **BTTrace**
- GetLogLevel() : **BTTrace**
- GetLogSizeInBytes() : **BTTrace**
- GetLogSizeInEntries() : **BTTrace**
- GetWindowProc() : **ATL::BTWindow< BASE_CLASS >**

- i -

- Insert() : **BTTrace**
- InsertF() : **BTTrace**
- InsertV() : **BTTrace**

- l -

- LONG() : **ATL::BTWindow< BASE_CLASS >** , **MFC::BTWindow< BASE_CLASS >** , **ATL::BTDialog< BASE_CLASS >**

- o -

- Open() : **BTTrace**
- operator=() : **BTTrace**

- p -

- PrivDialogProc() : **ATL::BTDialog< BASE_CLASS >**
- PrivWindowProc() : **MFC::BTWindow< BASE_CLASS >** ,
ATL::BTWindow< BASE_CLASS >

- s -

- SetDefaultLogLevel() : **BTTrace**
- SetLogEchoMode() : **BTTrace**
- SetLogFlags() : **BTTrace**
- SetLogLevel() : **BTTrace**
- SetLogSizeInBytes() : **BTTrace**

- SetLogSizeInEntries() : **BTTrace**

- ~ -

- ~BTTrace() : **BTTrace**

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1



- m -

- m_eDefaultLogLevel : **BTTrace**
- m_iHandle : **BTTrace**
- m_pfnBaseDlgProc : **ATL::BTDialog< BASE_CLASS >**
- m_pfnBaseWndProc : **ATL::BTWindow< BASE_CLASS >**

- p -

- param2 : **ATL::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **ATL::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**
- param3 : **ATL::BTWindow< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **ATL::BTWindow< BASE_CLASS >**
- param4 : **ATL::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**

- S -

- szLogFileName : **BUGTRAP_LOGFILEENTRY_tag**, **BUGTRAP_REGEXPORTENTRY_tag**
- szRegKey : **BUGTRAP_REGEXPORTENTRY_tag**

- t -

- T2 : **ATL::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**
- T3 : **ATL::BTWindow< BASE_CLASS >**, **ATL::BTDialog< BASE_CLASS >**, **MFC::BTWindow< BASE_CLASS >**

- T4 : MFC::BTWindow< BASE_CLASS > , ATL::BTWindow< BASE_CLASS > , ATL::BTDialo g< BASE_CLASS >
-

Generated on Sun Dec 21 21:40:08 2008 for BugTrap by  1.5.7.1

The screenshot shows a software documentation interface with the following navigation structure:

- Main Page
- Modules
- Namespaces
- Classes
- Files

Under the "Files" tab, the following sub-navigation is visible:

- File List
- File Members

Under "File Members", the following categories are listed:

- All
- Functions
- Typedefs
- Enumerations
- Enumerator
- Defines

A search bar at the bottom left contains the letter "b".

- b -

- BT_AddLogFile() : [BugTrap.h](#)
- BT_AddRegFile() : [BugTrap.h](#)
- BT_AppLogEntry() : [BugTrap.h](#)
- BT_AppLogEntryF() : [BugTrap.h](#)
- BT_AppLogEntryV() : [BugTrap.h](#)
- BT_CallCppFilter() : [BugTrap.h](#)
- BT_CallSehFilter() : [BugTrap.h](#)
- BT_ClearLog() : [BugTrap.h](#)
- BT_ClearLogFiles() : [BugTrap.h](#)
- BT_CloseLogFile() : [BugTrap.h](#)
- BT_CppFilter() : [BugTrap.h](#)
- BT_DeleteLogFile() : [BugTrap.h](#)
- BT_ExportRegistryKey() : [BugTrap.h](#)
- BT_FlushLogFile() : [BugTrap.h](#)
- BT_GetActivityType() : [BugTrap.h](#)
- BT_GetAppName() : [BugTrap.h](#)
- BT_GetAppVersion() : [BugTrap.h](#)
- BT_GetDialogMessage() : [BugTrap.h](#)
- BT_GetDumpType() : [BugTrap.h](#)
- BT_GetExitMode() : [BugTrap.h](#)
- BT_GetFlags() : [BugTrap.h](#)
- BT_GetLogEchoMode() : [BugTrap.h](#)
- BT_GetLogFileEntry() : [BugTrap.h](#)
- BT_GetLogFileName() : [BugTrap.h](#)
- BT_GetLogFilesCount() : [BugTrap.h](#)
- BT_GetLogFlags() : [BugTrap.h](#)
- BT_GetLogLevel() : [BugTrap.h](#)
- BT_GetLogSizeInBytes() : [BugTrap.h](#)
- BT_GetLogSizeInEntries() : [BugTrap.h](#)

- BT_GetMailProfile() : **BugTrap.h**
- BT_GetNotificationEMail() : **BugTrap.h**
- BT_GetPostErrHandler() : **BugTrap.h**
- BT_GetPreErrHandler() : **BugTrap.h**
- BT_GetReportFilePath() : **BugTrap.h**
- BT_GetReportFormat() : **BugTrap.h**
- BT_GetSupportEMail() : **BugTrap.h**
- BT_GetSupportHost() : **BugTrap.h**
- BT_GetSupportPort() : **BugTrap.h**
- BT_GetSupportURL() : **BugTrap.h**
- BT GetUserMessage() : **BugTrap.h**
- BT_InsLogEntry() : **BugTrap.h**
- BT_InsLogEntryF() : **BugTrap.h**
- BT_InsLogEntryV() : **BugTrap.h**
- BT_InstallSehFilter() : **BugTrap.h**
- BT_InterceptSUEF() : **BugTrap.h**
- BT_MailSnapshot() : **BugTrap.h**
- BT_MailSnapshotEx() : **BugTrap.h**
- BT_OpenLogFile() : **BugTrap.h**
- BT_ReadVersionInfo() : **BugTrap.h**
- BT_SaveSnapshot() : **BugTrap.h**
- BT_SaveSnapshotEx() : **BugTrap.h**
- BT_SehFilter() : **BugTrap.h**
- BT_SendSnapshot() : **BugTrap.h**
- BT_SendSnapshotEx() : **BugTrap.h**
- BT_SetActivityType() : **BugTrap.h**
- BT_SetAppName() : **BugTrap.h**
- BT_SetAppVersion() : **BugTrap.h**
- BT_SetDialogMessage() : **BugTrap.h**
- BT_SetDumpType() : **BugTrap.h**
- BT_SetExitMode() : **BugTrap.h**
- BT_SetFlags() : **BugTrap.h**
- BT_SetLogEchoMode() : **BugTrap.h**
- BT_SetLogFlags() : **BugTrap.h**
- BT_SetLogLevel() : **BugTrap.h**
- BT_SetLogSizeInBytes() : **BugTrap.h**
- BT_SetLogSizeInEntries() : **BugTrap.h**

- `BT_SetMailProfile()` : **BugTrap.h**
 - `BT_SetNotificationEMail()` : **BugTrap.h**
 - `BT_SetPostErrHandler()` : **BugTrap.h**
 - `BT_SetPreErrHandler()` : **BugTrap.h**
 - `BT_SetReportFilePath()` : **BugTrap.h**
 - `BT_SetReportFormat()` : **BugTrap.h**
 - `BT_SetSupportEMail()` : **BugTrap.h**
 - `BT_SetSupportHost()` : **BugTrap.h**
 - `BT_SetSupportPort()` : **BugTrap.h**
 - `BT_SetSupportServer()` : **BugTrap.h**
 - `BT_SetSupportURL()` : **BugTrap.h**
 - `BT_SetUserMessage()` : **BugTrap.h**
 - `BT_SetUserMessageFromCode()` : **BugTrap.h**
 - `BT_UninstallSehFilter()` : **BugTrap.h**
-

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

- BT_ErrHandler : [BugTrap.h](#)
- BUGTRAP_ACTIVITY : [BugTrap.h](#)
- BUGTRAP_DIALOGMESSAGE : [BugTrap.h](#)
- BUGTRAP_EXITMODE : [BugTrap.h](#)
- BUGTRAP_FLAGS : [BugTrap.h](#)
- BUGTRAP_LOGECHOTYPE : [BugTrap.h](#)
- BUGTRAP_LOGFILEENTRY : [BugTrap.h](#)
- BUGTRAP_LOGFLAGS : [BugTrap.h](#)
- BUGTRAP_LOGFORMAT : [BugTrap.h](#)
- BUGTRAP_LOGLEVEL : [BugTrap.h](#)
- BUGTRAP_LOGTYPE : [BugTrap.h](#)
- BUGTRAP_REGEXPORTENTRY : [BugTrap.h](#)
- BUGTRAP_REPORTFORMAT : [BugTrap.h](#)

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[File List](#)[File Members](#)[All](#)[Functions](#)[Typedefs](#)[Enumerations](#)[Enumerator](#)[Defines](#)

- BUGTRAP_ACTIVITY_tag : [BugTrap.h](#)
- BUGTRAP_DIALOGMESSAGE_tag : [BugTrap.h](#)
- BUGTRAP_EXITMODE_tag : [BugTrap.h](#)
- BUGTRAP_FLAGS_tag : [BugTrap.h](#)
- BUGTRAP_LOGECHOTYPE_tag : [BugTrap.h](#)
- BUGTRAP_LOGFLAGS_tag : [BugTrap.h](#)
- BUGTRAP_LOGFORMAT_tag : [BugTrap.h](#)
- BUGTRAP_LOGLEVEL_tag : [BugTrap.h](#)
- BUGTRAP_LOGTYPE_tag : [BugTrap.h](#)
- BUGTRAP_REPORTFORMAT_tag : [BugTrap.h](#)

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

>Main Page Modules Namespaces Classes Files

File List File Members

All Functions Typedefs Enumerations Enumerator Defines

b

- b -

- BTA_MAILREPORT : [BugTrap.h](#)
- BTA_SAVEREPORT : [BugTrap.h](#)
- BTA_SENDREPORT : [BugTrap.h](#)
- BTA_SHOWUI : [BugTrap.h](#)
- BTDM_INTRO1 : [BugTrap.h](#)
- BTDM_INTRO2 : [BugTrap.h](#)
- BTEM_CONTINUESEARCH : [BugTrap.h](#)
- BTEM_EXECUTEHANDLER : [BugTrap.h](#)
- BTEM_TERMINATEAPP : [BugTrap.h](#)
- BTF_ATTACHREPORT : [BugTrap.h](#)
- BTF_DETAILEDMODE : [BugTrap.h](#)
- BTF_EDITMAIL : [BugTrap.h](#)
- BTF_INTERCEPTSUEF : [BugTrap.h](#)
- BTF_LISTPROCESSES : [BugTrap.h](#)
- BTF_NONE : [BugTrap.h](#)
- BTF_SCREENCAPTURE : [BugTrap.h](#)
- BTF_SHOWADVANCEDUI : [BugTrap.h](#)
- BTLE_DBGOUT : [BugTrap.h](#)
- BTLE_NONE : [BugTrap.h](#)
- BTLE_STDERR : [BugTrap.h](#)
- BTLE_STDOUT : [BugTrap.h](#)
- BTLF_NONE : [BugTrap.h](#)
- BTLF_SHOWLOGLEVEL : [BugTrap.h](#)
- BTLF_SHOWTIMESTAMP : [BugTrap.h](#)
- BTLF_STREAM : [BugTrap.h](#)
- BTLF_TEXT : [BugTrap.h](#)
- BTLF_XML : [BugTrap.h](#)
- BTLL_ALL : [BugTrap.h](#)
- BTLL_ERROR : [BugTrap.h](#)

- BTLL_IMPORTANT : [BugTrap.h](#)
 - BTLL_INFO : [BugTrap.h](#)
 - BTLL_NONE : [BugTrap.h](#)
 - BTLL_VERBOSE : [BugTrap.h](#)
 - BTLL_WARNING : [BugTrap.h](#)
 - BTLT_LOGFILE : [BugTrap.h](#)
 - BTLT_REGEXPORT : [BugTrap.h](#)
 - BTRF_TEXT : [BugTrap.h](#)
 - BTRF_XML : [BugTrap.h](#)
-

Generated on Sun Dec 21 21:40:09 2008 for BugTrap by  1.5.7.1

[Main Page](#)[Modules](#)[Namespaces](#)[Classes](#)[Files](#)[File List](#)[File Members](#)[All](#)[Functions](#)[Typedefs](#)[Enumerations](#)[Enumerator](#)[Defines](#)

- `_BTDLG_INITIALIZER_` : [**BTAtlWindow.h**](#)
- `_BTWND_INITIALIZER_` : [**BTMfcWindow.h**](#) , [**BTAtlWindow.h**](#)
- `BT_DO_NOT_USE_DEFAULT_NAMESPACES` : [**BugTrap.h**](#)
- `BT_SetTerminate` : [**BugTrap.h**](#)
- `BUGTRAP_API` : [**BugTrap.h**](#)
- `BUGTRAP_HTTP_PORT` : [**BugTrap.h**](#)
- `MiniDumpNoDump` : [**BugTrap.h**](#)