

# IsoSDK Documentation

Version 10.2.0

# Table of Contents

## Symbol Reference

<b>Classes</b>	<b>1</b>
SAudioGrabbingParams Structure	2
SAudioGrabbingParams Members	2
SAudioGrabbingParams::SAudioGrabbingParams Constructor	3
SAudioGrabbingParams Data Members	3
SAudioGrabbingParams::nBitrate Data Member	4
SAudioGrabbingParams::nBitrateType Data Member	4
SAudioGrabbingParams::nEncoderType Data Member	5
SAudioGrabbingParams::nMaxBitrate Data Member	5
SAudioGrabbingParams::nMinBitrate Data Member	5
SAudioGrabbingParams::nNetworkTagsHandle Data Member	5
SAudioGrabbingParams::nQuality Data Member	5
SAudioGrabbingParams::nTagChoice Data Member	6
SAudioGrabbingParams Methods	6
SAudioGrabbingParams::INSERT_STRUCTURE_PADDING Method	6
SBootInfoEx Structure	6
SBootInfoEx Members	7
SBootInfoEx::SBootInfoEx Constructor	7
SBootInfoEx Data Members	7
SBootInfoEx::BootIndicator Data Member	8
SBootInfoEx::DeveloperID Data Member	8
SBootInfoEx::Emulation Data Member	8
SBootInfoEx::LoadSegment Data Member	9
SBootInfoEx::PlatformID Data Member	9
SBootInfoEx::SectorCount Data Member	9
SBootInfoEx Methods	9
SBootInfoEx::INSERT_STRUCTURE_PADDING Method	9
SBootVolumeInfo Structure	10
SBootVolumeInfo Members	10
SBootVolumeInfo::SBootVolumeInfo Constructor	10
SBootVolumeInfo Data Members	10
SBootVolumeInfo::nVolumeDescriptorAddress Data Member	11
SBootVolumeInfo::sInfoEx Data Member	11
SBootVolumeInfo Methods	11
SBootVolumeInfo::INSERT_STRUCTURE_PADDING Method	11
SCompressEncryptInfo Structure	11
SCompressEncryptInfo Members	12

SCompressEncryptInfo::SCompressEncryptInfo Constructor	12
SCompressEncryptInfo Data Members	12
SCompressEncryptInfo::bCompression Data Member	13
SCompressEncryptInfo::bEncryption Data Member	13
SCompressEncryptInfo::lpszPassword Data Member	13
SCompressEncryptInfo::nCompressionLevel Data Member	13
SCompressEncryptInfo Methods	13
SCompressEncryptInfo::INSERT_STRUCTURE_PADDING Method	13
SCreateImageParams Structure	14
SCreateImageParams Members	14
SCreateImageParams::SCreateImageParams Constructor	14
SCreateImageParams Data Members	15
SCreateImageParams::bFullCapacity Data Member	15
SCreateImageParams::cErrorParams Data Member	15
SCreateImageParams::lpszBadSectorsFilePath Data Member	15
SCreateImageParams::lpszImagePath Data Member	15
SCreateImageParams::nImageType Data Member	16
SCreateImageParams::nVerifyBufferSectors Data Member	16
SCreateImageParams Methods	16
SCreateImageParams::INSERT_STRUCTURE_PADDING Method	16
SDirToAdd Structure	16
SDirToAdd Members	17
SDirToAdd::SDirToAdd Constructor	17
SDirToAdd Data Members	17
SDirToAdd::bRecursive Data Member	18
SDirToAdd::lpszDestinationPath Data Member	18
SDirToAdd::lpszFileSpecification Data Member	18
SDirToAdd::lpszSourceDirPath Data Member	18
SDirToAdd::nFileAttributes Data Member	19
SDirToAdd::nSavePath Data Member	19
SDirToAdd Methods	19
SDirToAdd::INSERT_STRUCTURE_PADDING Method	19
SDirToCreate Structure	20
SDirToCreate Members	20
SDirToCreate::SDirToCreate Constructor	20
SDirToCreate Data Members	20
SDirToCreate::lpszDestinationPath Data Member	21
SDirToCreate::lpszDir Data Member	21
SDirToCreate Methods	21
SDirToCreate::INSERT_STRUCTURE_PADDING Method	21
SDirToRemove Structure	21
SDirToRemove Members	21

SDirToRemove::SDirToRemove Constructor	22
SDirToRemove Data Members	22
SDirToRemove::lpszDestinationPath Data Member	22
SDirToRemove Methods	22
SDirToRemove::INSERT_STRUCTURE_PADDING Method	22
SDirToRename Structure	23
SDirToRename Members	23
SDirToRename::SDirToRename Constructor	23
SDirToRename Data Members	23
SDirToRename::lpszDestinationPath Data Member	24
SDirToRename::lpszSourcePath Data Member	24
SDirToRename Methods	24
SDirToRename::INSERT_STRUCTURE_PADDING Method	24
SDiskCopyOptions Structure	24
SDiskCopyOptions Members	25
SDiskCopyOptions::SDiskCopyOptions Constructor	25
SDiskCopyOptions Data Members	25
SDiskCopyOptions::bEjectAfterBurn Data Member	26
SDiskCopyOptions::bFullCapacity Data Member	26
SDiskCopyOptions::bVerifyAfterBurn Data Member	26
SDiskCopyOptions::cErrorParams Data Member	27
SDiskCopyOptions::nReadMode Data Member	27
SDiskCopyOptions::nVerifyBufferSectors Data Member	27
SDiskCopyOptions::nWriteMethod Data Member	27
SDiskCopyOptions Methods	27
SDiskCopyOptions::INSERT_STRUCTURE_PADDING Method	27
SDVDVideoOptions Structure	28
SDVDVideoOptions Members	28
SDVDVideoOptions::SDVDVideoOptions Constructor	28
SDVDVideoOptions Data Members	28
SDVDVideoOptions::ForceUppercase Data Member	29
SDVDVideoOptions::Padding Data Member	29
SDVDVideoOptions Methods	29
SDVDVideoOptions::INSERT_STRUCTURE_PADDING Method	29
SExtendedDeviceInformation Structure	29
SExtendedDeviceInformation Members	30
SExtendedDeviceInformation::SExtendedDeviceInformation Constructor	31
SExtendedDeviceInformation Data Members	31
SExtendedDeviceInformation::FirmwareCreationDate Data Member	31
SExtendedDeviceInformation::lpszConnectionInterface Data Member	32
SExtendedDeviceInformation::lpszLoaderType Data Member	32
SExtendedDeviceInformation::lpszName Data Member	32

SExtendedDeviceInformation::lpszPhysicalInterface Data Member	32
SExtendedDeviceInformation::lpszRevision Data Member	32
SExtendedDeviceInformation::lpszSerialNumber Data Member	32
SExtendedDeviceInformation::nBufferSize Data Member	33
SExtendedDeviceInformation::nDeTransferMode Data Member	33
SExtendedDeviceInformation::nNumberOfVolumeLevels Data Member	33
SExtendedDeviceInformation::nReadRetryCount Data Member	33
SExtendedDeviceInformation::nRegionCode Data Member	33
SExtendedDeviceInformation::nRegionCodeChangesLeft Data Member	33
SExtendedDeviceInformation::nRegionCodeVendorResetsLeft Data Member	34
SExtendedDeviceInformation Methods	34
SExtendedDeviceInformation::INSERT_STRUCTURE_PADDING Method	34
SFileAudioProperty Structure	34
SFileAudioProperty Members	35
SFileAudioProperty::SFileAudioProperty Constructor	35
SFileAudioProperty Data Members	35
SFileAudioProperty::Arranger Data Member	36
SFileAudioProperty::bPauseInFrames Data Member	36
SFileAudioProperty::Composer Data Member	36
SFileAudioProperty::lpszMCN_ISRC Data Member	36
SFileAudioProperty::lpszSourceFilePath Data Member	36
SFileAudioProperty::lpszTitle Data Member	37
SFileAudioProperty::Message Data Member	37
SFileAudioProperty::nIndexesLength Data Member	37
SFileAudioProperty::nPause Data Member	37
SFileAudioProperty::Performer Data Member	37
SFileAudioProperty::pIndexes Data Member	37
SFileAudioProperty::SongWriter Data Member	38
SFileAudioProperty Methods	38
SFileAudioProperty::INSERT_STRUCTURE_PADDING Method	38
SFileDateTime Structure	38
SFileDateTime Members	38
SFileDateTime Data Members	39
SFileDateTime::nDay Data Member	39
SFileDateTime::nHour Data Member	39
SFileDateTime::nMinute Data Member	39
SFileDateTime::nMonth Data Member	39
SFileDateTime::nSecond Data Member	40
SFileDateTime::nYear Data Member	40
SFileDateTime Methods	40
SFileDateTime::INSERT_STRUCTURE_PADDING Method	40
SFileEntry Structure	40

SFileEntry Members	41
SFileEntry Data Members	41
SFileEntry::cAccessTime Data Member	42
SFileEntry::cCreationTime Data Member	42
SFileEntry::cDateTime Data Member	42
SFileEntry::lpszFileName Data Member	42
SFileEntry::lpszFileOrigin Data Member	42
SFileEntry::lpszFilePath Data Member	43
SFileEntry::nAddress Data Member	43
SFileEntry::nAttrib Data Member	43
SFileEntry::nFileSize Data Member	43
SFileEntry::pUserParam Data Member	43
SFileEntry Methods	44
INSERT_STRUCTURE_PADDING Method	44
SFileTimeEx Structure	44
SFileTimeEx Members	45
SFileTimeEx::SFileTimeEx Constructor	45
SFileTimeEx Data Members	46
SFileTimeEx::CreationDateTime Data Member	46
SFileTimeEx::LastAccessDateTime Data Member	46
SFileTimeEx::ModificationDateTime Data Member	46
SFileTimeEx::UseCreationDateTime Data Member	46
SFileTimeEx::UseCustomTimes Data Member	47
SFileTimeEx::UseLastAccessDateTime Data Member	47
SFileTimeEx::UseModificationDateTime Data Member	47
SFileToAdd Structure	47
SFileToAdd Members	48
SFileToAdd::SFileToAdd Constructor	48
SFileToAdd Data Members	48
SFileToAdd::bVideoFile Data Member	49
SFileToAdd::lpszDestinationPath Data Member	49
SFileToAdd::lpszFileName Data Member	49
SFileToAdd::lpszSourceFilePath Data Member	49
SFileToAdd::nSavePath Data Member	49
SFileToAdd Methods	50
SFileToAdd::INSERT_STRUCTURE_PADDING Method	50
SFileToAddEx Structure	50
SFileToAddEx Members	50
SFileToAddEx::SFileToAddEx Constructor	51
SFileToAddEx Data Members	51
SFileToAddEx::lpszDestinationPath Data Member	51
SFileToAddEx::lpszFileName Data Member	52

SFileToAddEx::lpszSourceFilePath Data Member	52
SFileToAddEx::nFileOffset Data Member	52
SFileToAddEx::nSavePath Data Member	52
SFileToAddEx::nSegmentSize Data Member	52
SFileToAddEx Methods	52
SFileToAddEx::INSERT_STRUCTURE_PADDING Method	53
SFileToRemove Structure	53
SFileToRemove Members	53
SFileToRemove::SFileToRemove Constructor	53
SFileToRemove Data Members	54
SFileToRemove::lpszDestinationPath Data Member	54
SFileToRemove::lpszFile Data Member	54
SFileToRemove Methods	54
SFileToRemove::INSERT_STRUCTURE_PADDING Method	54
SFileToRename Structure	54
SFileToRename Members	55
SFileToRename::SFileToRename Constructor	55
SFileToRename Data Members	55
SFileToRename::lpszDestinationPath Data Member	55
SFileToRename::lpszSourcePath Data Member	56
SFileToRename Methods	56
SFileToRename::INSERT_STRUCTURE_PADDING Method	56
SISOInfoEx Structure	56
SISOInfoEx Members	57
SISOInfoEx::SISOInfoEx Constructor	58
SISOInfoEx Data Members	58
SISOInfoEx::ISOAbstractFileIdentifier Data Member	60
SISOInfoEx::ISOAddSuffix Data Member	60
SISOInfoEx::ISOAllowLongISO9660Names Data Member	60
SISOInfoEx::ISOAllowLongJolietNames Data Member	60
SISOInfoEx::ISOAllowLowercaseNames Data Member	60
SISOInfoEx::ISOAllowManyDirectories Data Member	61
SISOInfoEx::ISOApplicationIdentifier Data Member	61
SISOInfoEx::ISOBiblioidentifier Data Member	61
SISOInfoEx::ISOCopyrightFileIdentifier Data Member	61
SISOInfoEx::ISOCreationDateTime Data Member	61
SISOInfoEx::ISODataPreparerIdentifier Data Member	61
SISOInfoEx::ISOEffectiveDateTime Data Member	62
SISOInfoEx::ISOExpirationDateTime Data Member	62
SISOInfoEx::ISOLevel Data Member	62
SISOInfoEx::ISOModificationDateTime Data Member	62
SISOInfoEx::ISOPublisherIdentifier Data Member	62

SISOInfoEx::ISOSetIdentifier Data Member	62
SISOInfoEx::ISOSystemIdentifier Data Member	63
SISOInfoEx::ISOUseCreationDateTime Data Member	63
SISOInfoEx::ISOUseEffectiveDateTime Data Member	63
SISOInfoEx::ISOUseExpirationDateTime Data Member	63
SISOInfoEx::ISOUseModificationDateTime Data Member	63
SISOInfoEx Methods	64
SISOInfoEx::INSERT_STRUCTURE_PADDING Method	64
SISOVolumeInfo Structure	64
SISOVolumeInfo Members	64
SISOVolumeInfo::SISOVolumeInfo Constructor	65
SISOVolumeInfo Data Members	65
SISOVolumeInfo::chVolumeLabel Data Member	65
SISOVolumeInfo::nPathTableAddress Data Member	65
SISOVolumeInfo::nPathTableSize Data Member	66
SISOVolumeInfo::nRootAddress Data Member	66
SISOVolumeInfo::nVolumeDescriptorAddress Data Member	66
SISOVolumeInfo::nVolumeSize Data Member	66
SISOVolumeInfo::sInfoEx Data Member	66
SISOVolumeInfo::tRootDateTime Data Member	66
SISOVolumeInfo Methods	67
SISOVolumeInfo::INSERT_STRUCTURE_PADDING Method	67
SMediumInfo Structure	67
SMediumInfo Members	67
SMediumInfo::SMediumInfo Constructor	69
SMediumInfo Data Members	69
SMediumInfo::chMediumType Data Member	71
SMediumInfo::chUPCEANCode Data Member	71
SMediumInfo::chVendorID Data Member	71
SMediumInfo::dMediumFreeSize Data Member	72
SMediumInfo::dMediumSize Data Member	72
SMediumInfo::dMediumUsedSize Data Member	72
SMediumInfo::fMaxWriteSpeed Data Member	72
SMediumInfo::nExtendedMediumType Data Member	72
SMediumInfo::nFirstSession Data Member	73
SMediumInfo::nFirstTrack Data Member	73
SMediumInfo::nLastSession Data Member	73
SMediumInfo::nLastSessionStatus Data Member	73
SMediumInfo::nLastTrack Data Member	73
SMediumInfo::nMediumStatus Data Member	73
SMediumInfo::wMediumTypeCode Data Member	74
SMediumInfo Methods	75

SMediumInfo::INSERT_STRUCTURE_PADDING Method	75
SOptions Structure	75
SOptions Members	76
SOptions::SOptions Constructor	77
SOptions Data Members	78
SOptions::bAutoErase Data Member	79
SOptions::bBootable Data Member	79
SOptions::bEjectAfterBurn Data Member	79
SOptions::bFinalize Data Member	79
SOptions::bJolietFileSystem Data Member	79
SOptions::bPadDataTracks Data Member	80
SOptions::bPerformOPC Data Member	80
SOptions::bRockRidgeFileSystem Data Member	80
SOptions::bTestBurn Data Member	80
SOptions::bUnderrunProtection Data Member	80
SOptions::bVerifyAfterBurn Data Member	81
SOptions::chBootImage Data Member	81
SOptions::chVolumeLabel Data Member	81
SOptions::nCacheSize Data Member	81
SOptions::nCopies Data Member	81
SOptions::nWriteMethod Data Member	81
SOptions Methods	82
SOptions::Empty Method	82
SOptions::INSERT_STRUCTURE_PADDING Method	82
SRAWTrackToAdd Structure	82
SRAWTrackToAdd Members	83
SRAWTrackToAdd::SRAWTrackToAdd Constructor	83
SRAWTrackToAdd Data Members	83
SRAWTrackToAdd::nDataTypeMask Data Member	84
SRAWTrackToAdd::nFormat Data Member	84
SRAWTrackToAdd::nIgnoreDataMask Data Member	85
SRAWTrackToAdd::nIndex Data Member	85
SRAWTrackToAdd::nLength Data Member	85
SRAWTrackToAdd::nNumber Data Member	85
SRAWTrackToAdd::nOffset Data Member	85
SRAWTrackToAdd::nStartAddress Data Member	85
SRAWTrackToAdd Methods	86
SRAWTrackToAdd::INSERT_STRUCTURE_PADDING Method	86
SReadErrorCorrectionParams Structure	86
SReadErrorCorrectionParams Members	86
SReadErrorCorrectionParams::SReadErrorCorrectionParams Constructor	87
SReadErrorCorrectionParams Data Members	87

SReadErrorCorrectionParams::bBlankBadSectors Data Member	87
SReadErrorCorrectionParams::bErrorCorrection Data Member	87
SReadErrorCorrectionParams::nHardwareRetryCount Data Member	87
SReadErrorCorrectionParams::nSoftwareRetryCount Data Member	88
SReadErrorCorrectionParams Methods	88
SReadErrorCorrectionParams::INSERT_STRUCTURE_PADDING Method	88
SSessionInfo Structure	88
SSessionInfo Members	88
SSessionInfo::SSessionInfo Constructor	89
SSessionInfo Data Members	89
SSessionInfo::bLastSession Data Member	89
SSessionInfo::dSessionSize Data Member	89
SSessionInfo::dStartLBA Data Member	90
SSessionInfo::nFirstTrack Data Member	90
SSessionInfo::nLastTrack Data Member	90
SSessionInfo Methods	90
SSessionInfo::INSERT_STRUCTURE_PADDING Method	90
SSpeed Structure	90
SSpeed Members	91
SSpeed::SSpeed Constructor	91
SSpeed Data Members	91
SSpeed::fSpeed Data Member	91
SSpeed::nSpeedInKBPerSec Data Member	91
SSpeed Methods	92
SSpeed::INSERT_STRUCTURE_PADDING Method	92
STags Structure	92
STags Members	92
STags::STags Constructor	93
STags Data Members	93
STags::chAlbum Data Member	93
STags::chArtist Data Member	93
STags::chComment Data Member	93
STags::chTitle Data Member	94
STags Methods	94
STags::Empty Method	94
STrackInfo Structure	94
STrackInfo Members	94
STrackInfo::STrackInfo Constructor	95
STrackInfo Data Members	95
STrackInfo::nFileSystem Data Member	96
STrackInfo::nFormat Data Member	96
STrackInfo::nSessionNumber Data Member	96

STrackInfo::nSize Data Member	96
STrackInfo::nStartLBA Data Member	96
STrackInfo::nTrackNumber Data Member	97
STrackInfo Methods	97
STrackInfo::INSERT_STRUCTURE_PADDING Method	97
SUDFOptions Structure	97
SUDFOptions Members	97
SUDFOptions::SUDFOptions Constructor	98
SUDFOptions Data Members	98
SUDFOptions::AvchdDisc Data Member	98
SUDFOptions::PartitionType Data Member	99
SUDFOptions::Version Data Member	99
SUDFOptions::WriteFileStreams Data Member	99
SUDFOptions Methods	99
SUDFOptions::INSERT_STRUCTURE_PADDING Method	100
SUDFVolumeInfo Structure	100
SUDFVolumeInfo Members	100
SUDFVolumeInfo::SUDFVolumeInfo Constructor	101
SUDFVolumeInfo Data Members	101
SUDFVolumeInfo::chPreparer Data Member	102
SUDFVolumeInfo::chVolumeLabel Data Member	102
SUDFVolumeInfo::nDirCount Data Member	102
SUDFVolumeInfo::nFileCount Data Member	102
SUDFVolumeInfo::nFSDAddress Data Member	103
SUDFVolumeInfo::nLVDAccount Data Member	103
SUDFVolumeInfo::nMetadataAddress Data Member	103
SUDFVolumeInfo::nMVDSAddress Data Member	103
SUDFVolumeInfo::nPartitionAddress Data Member	103
SUDFVolumeInfo::nPartitionLength Data Member	103
SUDFVolumeInfo::nPartitionType Data Member	103
SUDFVolumeInfo::nPVDAddress Data Member	104
SUDFVolumeInfo::nRootAddress Data Member	104
SUDFVolumeInfo::nRootFEAddress Data Member	104
SUDFVolumeInfo::nRVDSAddress Data Member	104
SUDFVolumeInfo::nSparingAddress Data Member	104
SUDFVolumeInfo::nVATAddress Data Member	104
SUDFVolumeInfo::nVersion Data Member	105
SUDFVolumeInfo::tRecordingDateTime Data Member	105
SUDFVolumeInfo Methods	105
SUDFVolumeInfo::INSERT_STRUCTURE_PADDING Method	105
SVideoFormat Structure	105
SVideoFormat Members	106

SVideoFormat::SVideoFormat Constructor	106
SVideoFormat Data Members	106
SVideoFormat::bUseable Data Member	107
SVideoFormat::fFPS Data Member	107
SVideoFormat::nAspectRatio Data Member	107
SVideoFormat::nBitRate Data Member	107
SVideoFormat::nHeight Data Member	108
SVideoFormat::nPlaytime Data Member	108
SVideoFormat::nWidth Data Member	108
SVideoFormat Methods	108
SVideoFormat::INSERT_STRUCTURE_PADDING Method	108
<b>Functions</b>	<b>108</b>
Abort Function	118
AddBurnDevice Function	118
AddDir Function	118
AddFile Function	119
AddFileEx Function	120
AnalyseDeviceCapability Function	120
AudioFileStop Function	121
Burn Function	121
BurnDialog Function	121
BurnISO Function	122
CheckLicenseKey Function	123
CheckSignature Function	123
ClearAll Function	123
CloseCDTextHandle Function	124
CloseDevice Function	124
CloseDirectory Function	124
CloseDiskSession Function	125
CloseNetworkTagsHandle Function	125
CloseSession Function	126
ConvertSpeedFromKBPerSec Function	126
CopyDisk Function	127
CreateDir Function	127
CreateImage Function	127
CreateProject Function	128
DABurn Function	129
DABurnISOImage Function	129
DACheckSignature Function	130
DACloseDevice Function	130
DACloseSession Function	130
DAConvertSpeedFromKBPerSec Function	131

DACopyDisk Function	131
DACreateImage Function	132
DAEjectDevice Function	132
DAErase Function	133
DAGetBurnSpeed Function	133
DAGetDeviceCapabilities Function	134
DAGetDeviceCapabilitiesHandle Function	134
DAGetDeviceInformation Function	135
DAGetDeviceInformationEx Function	135
DAGetMaxBurnSpeed Function	136
DAGetMaxReadSpeed Function	136
DAGetMediumFreedBld Function	137
DAGetMediumInformation Function	137
DAGetPossibleBurnSpeeds Function	138
DAGetPossibleImageFormats Function	138
DAGetPossibleReadSpeeds Function	139
DAGetReadSpeed Function	139
DAGetSessionInformation Function	140
DAGetTrackFormatEx Function	140
DAGetTrackIndexes Function	141
DAGetTrackInformation Function	141
DAGetTrackISRC Function	142
DAGrabAudioTrack Function	142
DAImportFile Function	143
DAImportFileEx Function	143
DAIsDeviceReady Function	144
DALockMedium Function	144
DAOpenDiskSession Function	145
DAPlayAudioTrack Function	145
DAPrepare Function	146
DAReadCDText Function	146
DAReadFileContents Function	147
DAReadSectors Function	147
DASaveTrackToFile Function	148
DASetBurnSpeed Function	148
DASetReadSpeed Function	149
DASetRegionalCode Function	149
DAVerifyFile Function	150
DeInitALL Function	150
DeInitialize Function	151
DeleteProject Function	151
DirExists Function	151

---

EjectDevice Function	152
EnableImageDevice Function	152
EnableMCNDisabling Function	153
Erase Function	153
EraseDialog Function	154
EraseMpegByIndex Function	154
ForceDeInitialize Function	154
GetActiveDevicesCount Function	155
GetASPI Function	155
GetAudioFileSize Function	156
GetAudioTags Function	156
GetBootInfoEx Function	156
GetBootVolumeInformation Function	157
GetBurnDevice Function	157
GetBurnDevices Function	158
GetBurnDoneEventCallback Function	158
GetBurnFileEventCallback Function	159
GetBurnSpeed Function	159
GetCDTextDiskTagString Function	160
GetCDTextTrackTagString Function	160
GetCompareFilesForArrangementEventCallback Function	161
GetCompressEncrypt Function	161
GetDeviceCapabilities Function	162
GetDeviceCapabilitiesHandle Function	162
GetDeviceInformation Function	163
GetDeviceInformationEx Function	163
GetDevices Function	164
GetDVDVideoOptions Function	164
GetEraseDoneEventCallback Function	165
GetErrorDeviceName Function	165
GetFileAllocationTable Function	165
GetFileEntry Function	166
GetFileTimeEx Function	166
GetFinalizeEventCallback Function	167
GetImagePath Function	167
GetImageSize Function	168
GetInfoTextEventCallback Function	168
GetISOInfoEx Function	168
GetISOVolumeInformation Function	169
GetJobDoneEventCallback Function	169
GetLanguage Function	170
GetMaxBurnSpeed Function	170

---

GetMaxReadSpeed Function	171
GetMediumFreedBld Function	171
GetMediumInformation Function	172
GetMpegCount Function	172
GetNetworkDiskTagInt Function	172
GetNetworkDiskTagString Function	173
GetNetworkTrackTagInt Function	174
GetNetworkTrackTagString Function	174
GetNumberOfFiles Function	175
GetOptions Function	175
GetPlayTime Function	176
GetPossibleBurnSpeeds Function	176
GetPossibleImageFormats Function	177
GetPossibleReadSpeeds Function	177
GetPrecisePlayTime Function	178
GetProcessEventCallback Function	178
GetProjectType Function	178
GetRAWDataEventCallback Function	179
GetReadDevice Function	179
GetReadSpeed Function	180
GetSessionInformation Function	180
GetStartVerifyEventCallback Function	181
GetText Function	181
GetTmpPath Function	182
GetTrackFormatEx Function	182
GetTrackIndexes Function	183
GetTrackInformation Function	183
GetTrackISRC Function	184
GetUDFOptions Function	184
GetUDFOptionsEx Function	185
GetUDFVolumeInformation Function	185
GetVerify Function	185
GetVerifyDoneEventCallback Function	186
GetVerifyErrorEventCallback Function	186
GetVerifyFileEventCallback Function	187
GetVerifySectorEventCallback Function	187
GetWriteCDTextInUnicode Function	187
GrabAudioTrack Function	188
ImportFile Function	188
ImportFileEx Function	189
Initialize Function	189
IsDeviceReady Function	190

IsValidVideoTsFolder Function	190
LoadBassPlugin Function	191
LockMedium Function	191
MultiDeviceDialog Function	192
OpenDirectory Function	192
OpenDiskSession Function	193
PlayAudioFile Function	193
PlayAudioTrack Function	194
Prepare Function	194
ReadCDText Function	194
ReadDirectory Function	195
ReadFileContents Function	195
ReadSectors Function	196
ReleaseDeviceCapabilities Function	196
RemoveBurnDevice Function	197
RemoveDir Function	197
RemoveFile Function	198
RenameDir Function	198
RenameFile Function	198
RescanDevices Function	199
ResetCallbacks Function	199
SaveLogToFile Function	200
SaveOptionsToFile Function	200
SaveTrackToFile Function	201
SetAddFileEventCallback Function	201
SetASPI Function	202
SetAudioDecodeDoneEventCallback Function	203
SetAudioDecoderEventCallback Function	203
SetAudioFileProperty Function	204
SetBootInfoEx Function	204
SetBurnDevice Function	205
SetBurnDoneEventCallback Function	205
SetBurnFileEventCallback Function	206
SetBurnSpeed Function	206
SetCompareFilesForArrangementEventCallback Function	207
SetCompressEncrypt Function	208
SetCreateDirEventCallback Function	208
SetDVDVideoOptions Function	209
SetEraseDoneEventCallback Function	209
SetFileAttr Function	210
SetFileTimeEx Function	210
SetFileTimes Function	211

---

SetFileUserParam Function	211
SetFinalizeEventCallback Function	212
SetFXApp Function	212
SetGetTextEventCallback Function	213
SetIgnoreFileExist Function	213
SetImagePath Function	214
SetInfoTextEventCallback Function	214
SetISOInfoEx Function	215
SetJobDoneEventCallback Function	215
SetLanguage Function	216
SetOptions Function	217
SetOptionsFromFile Function	217
SetProcessEventCallback Function	217
SetRAWDataEventCallback Function	218
SetRAWStructure Function	219
SetReadDevice Function	219
SetReadSpeed Function	219
SetRegionalCode Function	220
SetRemoveFileEventCallback Function	221
SetStartVerifyEventCallback Function	221
SetTmpPath Function	222
SetUDFOptions Function	222
SetUDFOptionsEx Function	223
SetVCDKeyHandler Function	223
SetVCDTimeOutHandler Function	224
SetVerify Function	224
SetVerifyDoneEventCallback Function	224
SetVerifyErrorEventCallback Function	225
SetVerifyFileEventCallback Function	226
SetVerifySectorEventCallback Function	226
SetVideoScanDoneEventCallback Function	227
SetVideoScannerEventCallback Function	228
SetWriteCDTextInUnicode Function	228
StopMpegAction Function	229
TagsFromNetworkDialog Function	229
VerifyFile Function	230
<b>Structs, Records, Enums</b>	<b>230</b>
_FoxEntries Structure	231
_FoxIndecies Structure	231
SExtent Structure	231
SFileAllocationTable Structure	232
SUDFOptionsEx Structure	232

FoxEntries Structure	233
FoxIndecies Structure	233
PFoxEntries Structure	234
PFoxIndecies Structure	234
<b>Types</b>	<b>234</b>
AddFileEvent Type	235
AudioDecodeDoneEvent Type	236
AudioDecoderEvent Type	236
BurnDoneEvent Type	237
BurnFileEvent Type	237
CompareFilesForArrangementEvent Type	237
CreateDirEvent Type	238
EraseDoneEvent Type	238
FinalizeEvent Type	239
GetTextEvent Type	239
HDIR Type	239
HSESSION Type	240
InfoTextEvent Type	240
int16 Type	240
int32 Type	241
int64 Type	241
int8 Type	241
JobDoneEvent Type	241
ProcessEvent Type	242
RAWDataEvent Type	242
RemoveFileEvent Type	243
StartVerifyEvent Type	243
uint16 Type	243
uint32 Type	244
uint8 Type	244
VerifyDoneEvent Type	244
VerifyErrorEvent Type	245
VerifyFileEvent Type	245
VerifySectorEvent Type	245
VideoScanDoneEvent Type	246
VideoScannerEvent Type	246
<b>Macros</b>	<b>247</b>
____stat64 Macro	305
__BS_LIBRARY_H__ Macro	305
__ISOSDK_DEFINITIONS_H__ Macro	306
__ISOSDK_EXPORT_H__ Macro	306

---

__max Macro	306
__min Macro	306
_FindData_t Macro	307
_FindFirst Macro	307
_MAX_PATH Macro	307
BOOL2bool Macro	307
BS_ANALOG_AUDIO_PLAYBACK Macro	308
BS_API Macro	308
BS_AR_16TO9_DISPLAY Macro	308
BS_AR_221TO2_DISPLAY Macro	308
BS_AR_4TO3_DISPLAY Macro	309
BS_AR_SQUARE_PIXELS Macro	309
BS_AR_UNKNOWN Macro	309
BS_ASPI_FROGASPI Macro	309
BS_ASPI_INTERNAL Macro	310
BS_ASPI_WNASPI Macro	310
BS_AUDIO_MP3 Macro	310
BS_AUDIO_NO Macro	310
BS_AUDIO_OGG Macro	311
BS_AUDIO_PCM Macro	311
BS_BARCODE_READ Macro	311
BS_BLURAY_R Macro	311
BS_BLURAY_R_RRM Macro	312
BS_BLURAY_RE Macro	312
BS_BLURAY_ROM Macro	312
BS_BOOL Macro	312
BS_BOOL2bool Macro	313
BS_BT_AVERAGE Macro	313
BS_BT_CONSTANT Macro	313
BS_BT_VARIABLE Macro	313
BS_BURN_DEVICE Macro	314
BS_BURNER_AUDIO Macro	314
BS_BURNER_BLURAY Macro	314
BS_BURNER_CUE Macro	314
BS_BURNER_DATA Macro	315
BS_BURNER_ISOUDF Macro	315
BS_BURNER_MIXEDMODE Macro	315
BS_BURNER_RAW Macro	315
BS_BURNER_SVCD Macro	316
BS_BURNER_UDFDVD Macro	316
BS_BURNER_VCD Macro	316
BS_BURNER_VIDEOODVD Macro	316

---

---

BS_C2_POINTERS Macro	317
BS_CALL Macro	317
BS_CD_R Macro	317
BS_CD_ROM Macro	317
BS_CD_RW Macro	318
BS_CD_TEXT_READ Macro	318
BS_CD_TEXT_WRITE Macro	318
BS_CDDA_COMMANDS Macro	318
BS_CDDA_STREAM_IS_ACCURATE Macro	319
BS_CDTCI_ARRANGER Macro	319
BS_CDTCI_COMPOSER Macro	319
BS_CDTCI_MESSAGE Macro	319
BS_CDTCI_PERFORMER Macro	320
BS_CDTCI_SONG_WRITER Macro	320
BS_CDTCI_TITLE Macro	320
BS_CHANGER_SIDE_CHANGE_CAPABLE Macro	320
BS_CHANGER_SOFTWARE_SLOT_SELECTION Macro	321
BS_CHANGER_SUPPORTS_DISC_PRESENT Macro	321
BS_COMPOSITE_AUDIO_AND_VIDEO Macro	321
BS_COMPRESSED_SIGNATURE Macro	321
BS_COMPRESSED_SIGNATURE_FOR_DRIVER Macro	322
BS_CONTINUE_LAST_SESSION Macro	322
BS_CONTINUE_NO_SESSION Macro	322
BS_CPRMAUTH Macro	322
BS_CURRENT_DEVICE Macro	323
BS.DAO_16 Macro	323
BS.DAO_96_PACK Macro	323
BS.DAO_96_RAW Macro	323
BS.DAO_RAW Macro	324
BS_DD_CD_R Macro	324
BS_DD_CD_ROM Macro	324
BS_DD_CD_RW Macro	324
BS_DEFAULT_LANGUAGE_FILE Macro	325
BS_DEFAULT_REGISTRY_KEY Macro	325
BS_DEFAULT_REGISTRY_PATH Macro	325
BS_DEFECTMANAGEMENT Macro	325
BS_DIGITAL_PORT_1 Macro	326
BS_DIGITAL_PORT_2 Macro	326
BS_DONT_SAVE_PATH Macro	326
BS_DVD_MRDL Macro	326
BS_DVD_PLUSR Macro	327
BS_DVD_PLUSRW Macro	327

---

---

BS_DVD_R Macro	327
BS_DVD_RAM Macro	327
BS_DVD_RDL_PLUS Macro	328
BS_DVD_ROM Macro	328
BS_DVD_RW Macro	328
BS_DVD_RW_RO Macro	328
BS_DVD_RW_SR Macro	329
BS_DVD_RWDL_PLUS Macro	329
BS_EJECT_INDIVIDUAL_OR_MAGAZINE Macro	329
BS_EMT_BD Macro	329
BS_EMT_CD_AUDIO Macro	330
BS_EMT_CD_ENHANCED Macro	330
BS_EMT_CD_MIXED_MODE Macro	330
BS_EMT_CD_MULTISESSION Macro	330
BS_EMT_CD_ROM Macro	331
BS_EMT_CD_ROM_XA Macro	331
BS_EMT_DVD Macro	331
BS_EMT_HDDVD Macro	331
BS_ET_AAC Macro	332
BS_ET_FLAC Macro	332
BS_ET_MP3 Macro	332
BS_ET_MP4 Macro	332
BS_ET_OGG Macro	333
BS_ET_OPUS Macro	333
BS_ET_WMA Macro	333
BS_FA_ADVANCED_HIDDEN Macro	333
BS_FA_ALL Macro	334
BS_FA_ARCHIVE Macro	334
BS_FA_DIRECTORY Macro	334
BS_FA_HIDDEN Macro	334
BS_FA_READONLY Macro	335
BS_FA_SYSTEM Macro	335
BS_FALSE Macro	335
BS_FF_BIN Macro	335
BS_FF_ISO Macro	336
BS_FF_MPEG Macro	336
BS_FF_WAVE Macro	336
BS_FS_BOOTABLE Macro	336
BS_FS_ISO9660 Macro	337
BS_FS_JOLIET Macro	337
BS_FS_ROCKRIDGE Macro	337
BS_FS_UDF Macro	337

---

BS_FS_UNKNOWN Macro	338
BS_GUI_ERROR_01 Macro	338
BS_GUI_ERROR_02 Macro	338
BS_GUI_RESOURCE_01 Macro	338
BS_GUI_RESOURCE_02 Macro	339
BS_GUI_RESOURCE_03 Macro	339
BS_GUI_RESOURCE_04 Macro	339
BS_GUI_RESOURCE_05 Macro	339
BS_GUI_RESOURCE_06 Macro	340
BS_GUI_RESOURCE_07 Macro	340
BS_GUI_RESOURCE_08 Macro	340
BS_GUI_RESOURCE_09 Macro	340
BS_GUI_RESOURCE_10 Macro	341
BS_GUI_RESOURCE_11 Macro	341
BS_GUI_RESOURCE_12 Macro	341
BS_GUI_RESOURCE_13 Macro	342
BS_GUI_RESOURCE_14 Macro	342
BS_GUI_RESOURCE_15 Macro	342
BS_GUI_RESOURCE_16 Macro	342
BS_GUI_RESOURCE_17 Macro	343
BS_GUI_RESOURCE_18 Macro	343
BS_GUI_RESOURCE_19 Macro	343
BS_GUI_RESOURCE_20 Macro	343
BS_GUI_RESOURCE_21 Macro	344
BS_GUI_RESOURCE_22 Macro	344
BS_GUI_RESOURCE_23 Macro	344
BS_GUI_RESOURCE_24 Macro	345
BS_GUI_RESOURCE_25 Macro	345
BS_GUI_RESOURCE_26 Macro	345
BS_GUI_RESOURCE_27 Macro	345
BS_GUI_RESOURCE_28 Macro	346
BS_GUI_RESOURCE_29 Macro	346
BS_GUI_RESOURCE_30 Macro	346
BS_GUI_RESOURCE_31 Macro	346
BS_GUI_RESOURCE_32 Macro	347
BS_GUI_RESOURCE_33 Macro	347
BS_GUI_RESOURCE_34 Macro	347
BS_GUI_RESOURCE_35 Macro	347
BS_GUI_RESOURCE_36 Macro	348
BS_GUI_RESOURCE_37 Macro	348
BS_GUI_RESOURCE_38 Macro	348
BS_GUI_RESOURCE_39 Macro	349

---

BS_GUI_RESOURCE_40 Macro	349
BS_GUI_RESOURCE_41 Macro	349
BS_GUI_RESOURCE_42 Macro	349
BS_GUI_RESOURCE_43 Macro	350
BS_GUI_RESOURCE_44 Macro	350
BS_GUI_RESOURCE_45 Macro	350
BS_GUI_RESOURCE_46 Macro	350
BS_GUI_RESOURCE_47 Macro	351
BS_GUI_RESOURCE_48 Macro	351
BS_GUI_RESOURCE_49 Macro	351
BS_GUI_RESOURCE_50 Macro	352
BS_GUI_RESOURCE_51 Macro	352
BS_GUI_RESOURCE_52 Macro	352
BS_GUI_RESOURCE_53 Macro	352
BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLUMN Macro	353
BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_COLUMN Macro	353
BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLUMN Macro	353
BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABEL Macro	354
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE Macro	354
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN Macro	354
BS_HD_DVD_R Macro	355
BS_HD_DVD_R_DL Macro	355
BS_HD_DVD_RAM Macro	355
BS_HD_DVD_ROM Macro	355
BS_HD_DVD_RW Macro	356
BS_HD_DVD_RW_DL Macro	356
BS_IL_HIGH_DEBUG Macro	356
BS_IL_INFO Macro	356
BS_IL_LOW_DEBUG Macro	357
BS_IL_MEDIUM_DEBUG Macro	357
BS_ILE_TOO_BIG_FILE Macro	357
BS_ILE_TOO_LONG_DIRECTORY_NESTING Macro	357
BS_IMG_BIN Macro	358
BS_IMG_ISO Macro	358
BS_IMGTASK_CREATE Macro	358
BS_IMGTASK_VERIFY Macro	358
BS_IMPOPTS_COMMON Macro	359
BS_IMPOPTS_DECRYPT Macro	359
BS_IMPOPTS_EX Macro	359
BS_IMPOPTS_UNCOMPRESS Macro	359
BS_INVALID_HANDLE Macro	360
BS_INVALID_TAG_HANDLE Macro	360

---

BS_ISO_LEVEL_1 Macro	360
BS_ISO_LEVEL_2 Macro	360
BS_ISO_LEVEL_3 Macro	361
BS_ISO_LEVEL_ROMEO Macro	361
BS_ISRC_READ Macro	361
BS_LABELFLASH Macro	362
BS_LAYER_JUMP_RECORDING Macro	362
BS_LIGHTSCRIBE Macro	362
BS_LOCK_MEDIA Macro	362
BS_LOCK_STATE Macro	363
BS_LP_BOOL Macro	363
BS_LS_COMPLETE_SESSION Macro	363
BS_LS_DAMAGED_SESSION Macro	363
BS_LS_EMPTY_SESSION Macro	364
BS_LS_INCOMPLETE_SESSION Macro	364
BS_MAX_SPEED Macro	364
BS_METHOD_2_ADDRESSING_FIXED_PACKETS Macro	364
BS_MODE2_FORM1_READ Macro	365
BS_MODE2_FORM2_READ Macro	365
BS_MS_COMPLETE_DISK Macro	365
BS_MS_EMPTY_DISK Macro	365
BS_MS_INCOMPLETE_DISK Macro	366
BS_MS_OTHER Macro	366
BS_MULTI_ERROR_01 Macro	366
BS_MULTI_ERROR_02 Macro	366
BS_MULTISESSION Macro	367
BS_NDEF Macro	367
BS_PACKET_WRITE Macro	367
BS_PARENTDIR_ONLY Macro	367
BS_PREVENT_JUMPER Macro	368
BS_R_W_SUBCHANNELS_DEINT_AND_CORR Macro	368
BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ Macro	368
BS_R_W_SUBCHANNELS_READ Macro	368
BS_RDT_DATA Macro	369
BS_RDT_EDC_ECC Macro	369
BS_RDT_SUBCH_PQ Macro	369
BS_RDT_SUBCH_PW Macro	369
BS_RDT_SUBCH_RW Macro	370
BS_RDT_SUBHEADERS Macro	370
BS_RDT_SYNC_HEADER Macro	370
BS_READ_BLURAY_R Macro	370
BS_READ_BLURAY_R_XL Macro	371

---

BS_READ_BLURAY_RE Macro	371
BS_READ_BLURAY_RE_XL Macro	371
BS_READ_BLURAY_ROM Macro	371
BS_READ_CDR Macro	372
BS_READ_CDRW Macro	372
BS_READ_CDRW_CAV Macro	372
BS_READ_DEVICE Macro	372
BS_READ_DVD Macro	373
BS_READ_DVD_DL Macro	373
BS_READ_DVD_MRDL Macro	373
BS_READ_DVD_RDL_PLUS Macro	373
BS_READ_DVD_RWDL_PLUS Macro	374
BS_READ_DVDR Macro	374
BS_READ_DVDR_PLUS Macro	374
BS_READ_DVDRAM Macro	374
BS_READ_DVDRW Macro	375
BS_READ_DVDRW_PLUS Macro	375
BS_READ_HDDVD_R Macro	375
BS_READ_HDDVD_ROM Macro	375
BS_READ_HDDVD_RW Macro	376
BS_READ_MOUNT_RAINER Macro	376
BS_RM_RAW Macro	376
BS_RM_RAW_SUBCHANNEL Macro	376
BS_RM_USERDATA Macro	377
BS_RTF_AUDIO Macro	377
BS_RTF_MODE1 Macro	377
BS_RTF_MODE2_FORM1 Macro	377
BS_RTF_MODE2_FORM2 Macro	378
BS_RTF_MODE2_FORMLESS Macro	378
BS_SCSI_ERROR_001 Macro	378
BS_SCSI_ERROR_002 Macro	378
BS_SCSI_ERROR_004 Macro	379
BS_SCSI_ERROR_005 Macro	379
BS_SCSI_ERROR_006 Macro	379
BS_SCSI_ERROR_007 Macro	379
BS_SCSI_ERROR_008 Macro	380
BS_SCSI_ERROR_009 Macro	380
BS_SCSI_ERROR_01 Macro	380
BS_SCSI_ERROR_010 Macro	380
BS_SCSI_ERROR_011 Macro	381
BS_SCSI_ERROR_012 Macro	381
BS_SCSI_ERROR_013 Macro	381

BS_SCSI_ERROR_014 Macro	381
BS_SCSI_ERROR_015 Macro	382
BS_SCSI_ERROR_016 Macro	382
BS_SCSI_ERROR_017 Macro	382
BS_SCSI_ERROR_018 Macro	382
BS_SCSI_ERROR_019 Macro	383
BS_SCSI_ERROR_02 Macro	383
BS_SCSI_ERROR_020 Macro	383
BS_SCSI_ERROR_021 Macro	383
BS_SCSI_ERROR_022 Macro	384
BS_SCSI_ERROR_023 Macro	384
BS_SCSI_ERROR_024 Macro	384
BS_SCSI_ERROR_025 Macro	384
BS_SCSI_ERROR_026 Macro	385
BS_SCSI_ERROR_027 Macro	385
BS_SCSI_ERROR_028 Macro	385
BS_SCSI_ERROR_029 Macro	385
BS_SCSI_ERROR_030 Macro	386
BS_SCSI_ERROR_031 Macro	386
BS_SCSI_ERROR_032 Macro	386
BS_SCSI_ERROR_033 Macro	386
BS_SCSI_ERROR_034 Macro	387
BS_SCSI_ERROR_035 Macro	387
BS_SCSI_ERROR_036 Macro	387
BS_SCSI_ERROR_037 Macro	387
BS_SCSI_ERROR_038 Macro	388
BS_SCSI_ERROR_039 Macro	388
BS_SCSI_ERROR_040 Macro	388
BS_SCSI_ERROR_041 Macro	388
BS_SCSI_ERROR_042 Macro	389
BS_SCSI_ERROR_043 Macro	389
BS_SCSI_ERROR_044 Macro	389
BS_SCSI_ERROR_045 Macro	389
BS_SCSI_ERROR_046 Macro	390
BS_SCSI_ERROR_047 Macro	390
BS_SCSI_ERROR_048 Macro	390
BS_SCSI_ERROR_049 Macro	390
BS_SCSI_ERROR_050 Macro	391
BS_SCSI_ERROR_051 Macro	391
BS_SCSI_ERROR_052 Macro	391
BS_SCSI_ERROR_053 Macro	391
BS_SCSI_ERROR_054 Macro	392

BS_SCSI_ERROR_055 Macro	392
BS_SCSI_ERROR_056 Macro	392
BS_SCSI_ERROR_057 Macro	392
BS_SCSI_ERROR_058 Macro	393
BS_SCSI_ERROR_059 Macro	393
BS_SCSI_ERROR_060 Macro	393
BS_SCSI_ERROR_061 Macro	393
BS_SCSI_ERROR_062 Macro	394
BS_SCSI_ERROR_063 Macro	394
BS_SCSI_ERROR_064 Macro	394
BS_SCSI_ERROR_065 Macro	394
BS_SCSI_ERROR_066 Macro	395
BS_SCSI_ERROR_067 Macro	395
BS_SCSI_ERROR_068 Macro	395
BS_SCSI_ERROR_069 Macro	395
BS_SCSI_ERROR_070 Macro	396
BS_SCSI_ERROR_071 Macro	396
BS_SCSI_ERROR_072 Macro	396
BS_SCSI_ERROR_073 Macro	396
BS_SCSI_ERROR_074 Macro	397
BS_SCSI_ERROR_075 Macro	397
BS_SCSI_ERROR_076 Macro	397
BS_SCSI_ERROR_077 Macro	397
BS_SCSI_ERROR_078 Macro	398
BS_SCSI_ERROR_079 Macro	398
BS_SCSI_ERROR_080 Macro	398
BS_SCSI_ERROR_081 Macro	398
BS_SCSI_ERROR_082 Macro	399
BS_SCSI_ERROR_083 Macro	399
BS_SCSI_ERROR_084 Macro	399
BS_SCSI_ERROR_085 Macro	399
BS_SCSI_ERROR_086 Macro	400
BS_SCSI_ERROR_087 Macro	400
BS_SCSI_ERROR_088 Macro	400
BS_SCSI_ERROR_089 Macro	400
BS_SCSI_ERROR_090 Macro	401
BS_SCSI_ERROR_091 Macro	401
BS_SCSI_ERROR_092 Macro	401
BS_SCSI_ERROR_093 Macro	401
BS_SCSI_ERROR_094 Macro	402
BS_SCSI_ERROR_095 Macro	402
BS_SCSI_ERROR_096 Macro	402

BS_SCSI_ERROR_097 Macro	402
BS_SCSI_ERROR_098 Macro	403
BS_SCSI_ERROR_099 Macro	403
BS_SCSI_ERROR_100 Macro	403
BS_SCSI_ERROR_101 Macro	403
BS_SCSI_ERROR_102 Macro	404
BS_SCSI_ERROR_103 Macro	404
BS_SCSI_ERROR_104 Macro	404
BS_SCSI_ERROR_105 Macro	404
BS_SCSI_ERROR_106 Macro	405
BS_SCSI_ERROR_107 Macro	405
BS_SCSI_ERROR_108 Macro	405
BS_SCSI_ERROR_109 Macro	405
BS_SCSI_ERROR_110 Macro	406
BS_SCSI_ERROR_111 Macro	406
BS_SCSI_ERROR_112 Macro	406
BS_SCSI_ERROR_113 Macro	406
BS_SCSI_ERROR_114 Macro	407
BS_SCSI_ERROR_115 Macro	407
BS_SCSI_ERROR_116 Macro	407
BS_SCSI_ERROR_117 Macro	407
BS_SCSI_ERROR_118 Macro	408
BS_SCSI_ERROR_119 Macro	408
BS_SCSI_ERROR_ALLOC_01 Macro	408
BS_SCSI_ERROR_ALLOC_02 Macro	408
BS_SCSI_ERROR_ASPI_01 Macro	409
BS_SCSI_ERROR_ASPI_02 Macro	409
BS_SCSI_ERROR_ASPI_03 Macro	409
BS_SCSI_ERROR_ASPI_04 Macro	409
BS_SCSI_ERROR_ASPI_05 Macro	410
BS_SCSI_ERROR_ASPI_06 Macro	410
BS_SCSI_ERROR_ASPI_07 Macro	410
BS_SCSI_ERROR_ASPI_08 Macro	410
BS_SCSI_ERROR_ASPI_09 Macro	411
BS_SCSI_ERROR_ASPI_10 Macro	411
BS_SCSI_ERROR_ATT_01 Macro	411
BS_SCSI_ERROR_ATT_02 Macro	411
BS_SCSI_ERROR_ATT_03 Macro	412
BS_SCSI_ERROR_ATT_04 Macro	412
BS_SCSI_ERROR_ATT_05 Macro	412
BS_SCSI_ERROR_AUDIO_01 Macro	412
BS_SCSI_ERROR_AUDIO_02 Macro	413

BS_SCSI_ERROR_AUDIO_03 Macro	413
BS_SCSI_ERROR_AUDIO_04 Macro	413
BS_SCSI_ERROR_AUDIO_05 Macro	413
BS_SCSI_ERROR_CDB_01 Macro	414
BS_SCSI_ERROR_CDB_02 Macro	414
BS_SCSI_ERROR_CIRC_01 Macro	414
BS_SCSI_ERROR_COMMAND_02 Macro	414
BS_SCSI_ERROR_COMMAND_03 Macro	415
BS_SCSI_ERROR_COMMAND_04 Macro	415
BS_SCSI_ERROR_COMMAND_05 Macro	415
BS_SCSI_ERROR_CRC_01 Macro	415
BS_SCSI_ERROR_DCSS_01 Macro	416
BS_SCSI_ERROR_DCSS_02 Macro	416
BS_SCSI_ERROR_DCSS_03 Macro	416
BS_SCSI_ERROR_DCSS_04 Macro	416
BS_SCSI_ERROR_DCSS_05 Macro	417
BS_SCSI_ERROR_DCSS_06 Macro	417
BS_SCSI_ERROR_DCSS_07 Macro	417
BS_SCSI_ERROR_DCSS_08 Macro	417
BS_SCSI_ERROR_DECOM_01 Macro	418
BS_SCSI_ERROR_DISK_01 Macro	418
BS_SCSI_ERROR_DISK_02 Macro	418
BS_SCSI_ERROR_DISK_03 Macro	418
BS_SCSI_ERROR_DISK_04 Macro	419
BS_SCSI_ERROR_DISK_05 Macro	419
BS_SCSI_ERROR_DISK_06 Macro	419
BS_SCSI_ERROR_DISK_07 Macro	419
BS_SCSI_ERROR_DISK_08 Macro	420
BS_SCSI_ERROR_DISK_09 Macro	420
BS_SCSI_ERROR_DISK_10 Macro	420
BS_SCSI_ERROR_DISK_11 Macro	420
BS_SCSI_ERROR_DISK_12 Macro	421
BS_SCSI_ERROR_DISK_13 Macro	421
BS_SCSI_ERROR_DISK_14 Macro	421
BS_SCSI_ERROR_DISK_15 Macro	421
BS_SCSI_ERROR_DISK_16 Macro	422
BS_SCSI_ERROR_DISK_17 Macro	422
BS_SCSI_ERROR_DISK_18 Macro	422
BS_SCSI_ERROR_DISK_19 Macro	422
BS_SCSI_ERROR_DISK_20 Macro	423
BS_SCSI_ERROR_DISK_21 Macro	423
BS_SCSI_ERROR_DISK_22 Macro	423

BS_SCSI_ERROR_DISK_23 Macro	423
BS_SCSI_ERROR_DISK_24 Macro	424
BS_SCSI_ERROR_DISK_25 Macro	424
BS_SCSI_ERROR_DISK_26 Macro	424
BS_SCSI_ERROR_DISK_27 Macro	424
BS_SCSI_ERROR_DISK_28 Macro	425
BS_SCSI_ERROR_DISK_29 Macro	425
BS_SCSI_ERROR_DISK_30 Macro	425
BS_SCSI_ERROR_DISK_31 Macro	425
BS_SCSI_ERROR_DISK_32 Macro	426
BS_SCSI_ERROR_DISK_33 Macro	426
BS_SCSI_ERROR_DISK_34 Macro	426
BS_SCSI_ERROR_DISK_35 Macro	426
BS_SCSI_ERROR_DISK_36 Macro	427
BS_SCSI_ERROR_DISK_37 Macro	427
BS_SCSI_ERROR_DRIVE_01 Macro	427
BS_SCSI_ERROR_DRIVE_02 Macro	427
BS_SCSI_ERROR_EXT_01 Macro	428
BS_SCSI_ERROR_EXT_02 Macro	428
BS_SCSI_ERROR_LOG_01 Macro	428
BS_SCSI_ERROR_LOG_02 Macro	428
BS_SCSI_ERROR_LOG_03 Macro	429
BS_SCSI_ERROR_LOG_04 Macro	429
BS_SCSI_ERROR_MECH_01 Macro	429
BS_SCSI_ERROR_MECH_02 Macro	429
BS_SCSI_ERROR_MECH_03 Macro	430
BS_SCSI_ERROR_PARAM_01 Macro	430
BS_SCSI_ERROR_PARAM_02 Macro	430
BS_SCSI_ERROR_PARAM_03 Macro	430
BS_SCSI_ERROR_PARAM_04 Macro	431
BS_SCSI_ERROR_PARAM_05 Macro	431
BS_SCSI_ERROR_PARAM_06 Macro	431
BS_SCSI_ERROR_PARAM_07 Macro	431
BS_SCSI_ERROR_PARAM_08 Macro	432
BS_SCSI_ERROR_PARAM_09 Macro	432
BS_SCSI_ERROR_PARAM_10 Macro	432
BS_SCSI_ERROR_READ_01 Macro	432
BS_SCSI_ERROR_READ_02 Macro	433
BS_SCSI_ERROR_RECOVER_01 Macro	433
BS_SCSI_ERROR_RECOVER_02 Macro	433
BS_SCSI_ERROR_RECOVER_03 Macro	433
BS_SCSI_ERROR_RECOVER_04 Macro	434

BS_SCSI_ERROR_RECOVER_05 Macro	434
BS_SCSI_ERROR_RECOVER_06 Macro	434
BS_SCSI_ERROR_RECOVER_07 Macro	434
BS_SCSI_ERROR_RECOVER_08 Macro	435
BS_SCSI_ERROR_RECOVER_09 Macro	435
BS_SCSI_ERROR_RECOVER_10 Macro	435
BS_SCSI_ERROR_RECOVER_11 Macro	435
BS_SCSI_ERROR_RECOVER_12 Macro	436
BS_SCSI_ERROR_RECOVER_13 Macro	436
BS_SCSI_ERROR_RECOVER_14 Macro	436
BS_SCSI_ERROR_RECOVER_15 Macro	436
BS_SCSI_ERROR_RECOVER_16 Macro	437
BS_SCSI_ERROR_RECOVER_17 Macro	437
BS_SCSI_ERROR_SEGM_01 Macro	437
BS_SCSI_ERROR_SEGM_02 Macro	437
BS_SCSI_ERROR_SEGM_03 Macro	438
BS_SCSI_ERROR_SEGM_04 Macro	438
BS_SCSI_ERROR_SESSION_01 Macro	438
BS_SCSI_ERROR_SESSION_02 Macro	438
BS_SCSI_ERROR_SESSION_03 Macro	439
BS_SCSI_ERROR_SESSION_04 Macro	439
BS_SCSI_ERROR_TARGET_01 Macro	439
BS_SCSI_ERROR_TARGET_02 Macro	439
BS_SCSI_ERROR_TARGET_03 Macro	440
BS_SCSI_ERROR_TARGET_04 Macro	440
BS_SCSI_ERROR_UNIT_01 Macro	440
BS_SCSI_ERROR_UNIT_02 Macro	440
BS_SCSI_ERROR_UNIT_03 Macro	441
BS_SCSI_ERROR_UNIT_04 Macro	441
BS_SCSI_ERROR_UNIT_05 Macro	441
BS_SCSI_ERROR_UNIT_06 Macro	441
BS_SCSI_ERROR_UNIT_07 Macro	442
BS_SCSI_ERROR_UNIT_08 Macro	442
BS_SCSI_ERROR_UNIT_09 Macro	442
BS_SCSI_ERROR_UNIT_10 Macro	442
BS_SCSI_ERROR_UNIT_11 Macro	443
BS_SCSI_ERROR_UNIT_12 Macro	443
BS_SCSI_ERROR_UNIT_13 Macro	443
BS_SCSI_ERROR_UNIT_14 Macro	443
BS_SCSI_ERROR_UNIT_16 Macro	444
BS_SCSI_ERROR_UNIT_17 Macro	444
BS_SCSI_ERROR_UNIT_18 Macro	444

BS_SCSI_ERROR_UNIT_19 Macro	444
BS_SCSI_ERROR_UNIT_20 Macro	445
BS_SCSI_ERROR_UNIT_21 Macro	445
BS_SCSI_ERROR_UNIT_22 Macro	445
BS_SCSI_ERROR_UNIT_23 Macro	445
BS_SCSI_ERROR_VOL_01 Macro	446
BS_SCSI_ERROR_VOL_02 Macro	446
BS_SCSI_ERROR_VOL_03 Macro	446
BS_SCSI_ERROR_VOL_04 Macro	446
BS_SCSI_ERROR_WRITE_01 Macro	447
BS_SCSI_ERROR_WRITE_02 Macro	447
BS_SCSI_ERROR_WRITE_03 Macro	447
BS_SCSI_ERROR_WRITE_04 Macro	447
BS_SCSI_ERROR_WRITE_05 Macro	448
BS_SCSI_ERROR_WRITE_06 Macro	448
BS_SCSI_ERROR_WRITE_07 Macro	448
BS_SCSI_ERROR_WRITE_08 Macro	448
BS_SCSI_ERROR_WRITE_09 Macro	449
BS_SCSI_ERROR_WRITE_10 Macro	449
BS_SCSI_ERROR_WRITE_11 Macro	449
BS_SCSI_ERROR_WRITE_12 Macro	449
BS_SDK_COMPENC_BOTH Macro	450
BS_SDK_COMPENC_COMPRESSED Macro	450
BS_SDK_COMPENC_ENCRYPTED Macro	450
BS_SDK_CUE_ERROR_COMMAND_01 Macro	450
BS_SDK_CUE_ERROR_COMMAND_06 Macro	451
BS_SDK_CUE_ERROR_FIELD Macro	451
BS_SDK_CUE_ERROR_FILE Macro	451
BS_SDK_CUE_ERROR_SENDING_CUE Macro	451
BS_SDK_CUE_ERROR_UEOL Macro	452
BS_SDK_ERROR_ABORTED Macro	452
BS_SDK_ERROR_ABORTEDCOMMAND Macro	452
BS_SDK_ERROR_BAD_CAPABILITY_NAME Macro	452
BS_SDK_ERROR_BAD_REQUEST Macro	453
BS_SDK_ERROR_BIN_FILE_NOT_FOUND Macro	453
BS_SDK_ERROR_BUFFERALIGNMENT Macro	453
BS_SDK_ERROR_BUFFERTOOBIG Macro	453
BS_SDK_ERROR_BURN_IN_PROGRESS Macro	454
BS_SDK_ERROR_CDTEXT_NOT_FOUND Macro	454
BS_SDK_ERROR_CHECKCONDITION Macro	454
BS_SDK_ERROR_COMPENC_GENERALERROR Macro	454
BS_SDK_ERROR_COMPENC_READ10_IDENT Macro	455

BS_SDK_ERROR_COMPRESSION_CONFLICT Macro	455
BS_SDK_ERROR_COMPRESSION_NOINDXTABLE Macro	455
BS_SDK_ERROR_COMPRESSION_NOSIGNATURE Macro	455
BS_SDK_ERROR_COMPRESSION_READ10 Macro	456
BS_SDK_ERROR_COMPRESSION_READMEMF Macro	456
BS_SDK_ERROR_COMPRESSION_SEEKMEMFILE Macro	456
BS_SDK_ERROR_COPYABORTED Macro	457
BS_SDK_ERROR_CORRUPT_OR_INVALID_CUE_FILE Macro	457
BS_SDK_ERROR_CREATEFILE Macro	457
BS_SDK_ERROR_DATAOVERRUN Macro	457
BS_SDK_ERROR_DATAPROTECT Macro	458
BS_SDK_ERROR_DATARECOVERERROR Macro	458
BS_SDK_ERROR_DEVICE_LOCKED Macro	458
BS_SDK_ERROR_EMPTY_PASSWORD Macro	458
BS_SDK_ERROR_ENCLIB_NOT_FOUND Macro	459
BS_SDK_ERROR_ENCRYPTION_CONFLICT Macro	459
BS_SDK_ERROR_ERASECHECK Macro	459
BS_SDK_ERROR_ERRINVALIDFILENAME Macro	459
BS_SDK_ERROR_FILE_EXISTS Macro	460
BS_SDK_ERROR_FILE_OPEN Macro	460
BS_SDK_ERROR_FILEINUSE Macro	460
BS_SDK_ERROR_FILEMARK Macro	460
BS_SDK_ERROR_GENERAL Macro	461
BS_SDK_ERROR_HARDWAREERROR Macro	461
BS_SDK_ERROR_ILLEGALLength Macro	461
BS_SDK_ERROR_IMPORTSESSION Macro	461
BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE Macro	462
BS_SDK_ERROR_INCORRECTLENGTH Macro	462
BS_SDK_ERROR_INVALID_DEST_PATH Macro	462
BS_SDK_ERROR_INVALID_DIR_INDEX Macro	462
BS_SDK_ERROR_INVALID_FILE_FORMAT Macro	463
BS_SDK_ERROR_INVALID_FILE_NAME Macro	463
BS_SDK_ERROR_INVALID_HANDLE Macro	463
BS_SDK_ERROR_INVALID_INDEX Macro	463
BS_SDK_ERROR_INVALID_ISRC Macro	464
BS_SDK_ERROR_INVALID_MCN Macro	464
BS_SDK_ERROR_INVALID_PATH Macro	464
BS_SDK_ERROR_INVALID_SESSION_NUMBER Macro	464
BS_SDK_ERROR_INVALID_SRC_PATH Macro	465
BS_SDK_ERROR_INVALID_UDF_VERSION Macro	465
BS_SDK_ERROR_INVALIDSRB Macro	465
BS_SDK_ERROR_ISOIMAGENOTFOUND Macro	465

BS_SDK_ERROR_MAXDIRS Macro	466
BS_SDK_ERROR_MAXFILES Macro	466
BS_SDK_ERROR_MESSAGEJECT Macro	466
BS_SDK_ERROR_MISCOMPARE Macro	466
BS_SDK_ERROR_MORE_SPACE_NEEDED Macro	467
BS_SDK_ERROR_MP3LIB_NOT_FOUND Macro	467
BS_SDK_ERROR_NETTAGS_CONNECT Macro	467
BS_SDK_ERROR_NETTAGS_DISK Macro	467
BS_SDK_ERROR_NETTAGS_INTERNAL Macro	468
BS_SDK_ERROR_NETTAGS_NOMATCH Macro	468
BS_SDK_ERROR_NETTAGS_SERVER Macro	468
BS_SDK_ERROR_NEXTADDRESS Macro	468
BS_SDK_ERROR_NO Macro	469
BS_SDK_ERROR_NOT_ALLOWED Macro	469
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER Macro	469
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE Macro	469
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION Macro	470
BS_SDK_ERROR_NOT_IMPLEMENTED Macro	470
BS_SDK_ERROR_NOTREADY Macro	470
BS_SDK_ERROR_NOTSUPPORTED Macro	471
BS_SDK_ERROR_PARITYERR Macro	471
BS_SDK_ERROR_PATH_EXISTS Macro	471
BS_SDK_ERROR_PLUGIN Macro	471
BS_SDK_ERROR_RESERVED Macro	472
BS_SDK_ERROR_SELECTIONTIMEOUT Macro	472
BS_SDK_ERROR_SRBTIMEOUT Macro	472
BS_SDK_ERROR_TIMEOUT Macro	472
BS_SDK_ERROR_TOOMUCH_DATA Macro	473
BS_SDK_ERROR_TOOMUCH_INDEXES Macro	473
BS_SDK_ERROR_UNEXPECTEDBUSFREE Macro	473
BS_SDK_ERROR_UNITATTENTION Macro	473
BS_SDK_ERROR_UNKNOWN Macro	474
BS_SDK_ERROR_UNKNOWN_TEXTID Macro	474
BS_SDK_ERROR_UNSUPPORTED_MEDIUM Macro	474
BS_SDK_ERROR_VOLUME_OVERFLOW Macro	474
BS_SDK_INT_ERROR_1 Macro	475
BS_SDK_INT_ERROR_2 Macro	475
BS_SDK_INT_ERROR_3 Macro	475
BS_SDK_INT_ERROR_4 Macro	475
BS_SDK_INT_ERROR_5 Macro	476
BS_SDK_INT_ERROR_FORMAT Macro	476
BS_SDK_KEY_INVALID Macro	476

BS_SDK_KEY_VALID Macro	476
BS_SDK_MESSAGE_01 Macro	477
BS_SDK_MESSAGE_02 Macro	477
BS_SDK_MESSAGE_03 Macro	477
BS_SDK_MESSAGE_04 Macro	478
BS_SDK_MESSAGE_05 Macro	478
BS_SDK_MESSAGE_06 Macro	478
BS_SDK_MESSAGE_07 Macro	479
BS_SDK_MESSAGE_08 Macro	479
BS_SDK_MESSAGE_10 Macro	479
BS_SDK_MESSAGE_11 Macro	480
BS_SDK_MESSAGE_12 Macro	480
BS_SDK_MESSAGE_13 Macro	480
BS_SDK_MESSAGE_14 Macro	481
BS_SDK_MESSAGE_15 Macro	481
BS_SDK_MESSAGE_16 Macro	481
BS_SDK_MESSAGE_COMPENC_BOTH Macro	482
BS_SDK_MESSAGE_COMPENC_COMPRESSED Macro	482
BS_SDK_MESSAGE_COMPENC_ENCRYPTED Macro	482
BS_SDK_MESSAGE_ERASESTART Macro	482
BS_SDK_MESSAGE_EXTR_FILE Macro	483
BS_SDK_MESSAGE_FORMAT Macro	483
BS_SDK_MESSAGE_FORMAT_DONE Macro	483
BS_SDK_MESSAGE_IMAGECREATESTART Macro	484
BS_SDK_MESSAGE_IMPORT Macro	484
BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY Macro	484
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_MATCHING_PARAM Macro	485
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_PROVIDER Macro	485
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES Macro	486
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY Macro	486
BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY Macro	487
BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES Macro	487
BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS Macro	488
BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE Macro	488
BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES Macro	489
BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY Macro	489
BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH Macro	490
BS_SDK_MESSAGE_SIMULATE Macro	490
BS_SDK_MESSAGE_WAIT Macro	490
BS_SDK_MESSAGE_WRIESTART Macro	491
BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE Macro	491
BS_SDK_VERIFY_ERROR_FILESDIFFERENT Macro	491

BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE Macro	491
BS_SEPARATE_CHANNEL_MUTE Macro	492
BS_SEPARATE_VOLUME_LEVELS Macro	492
BS_SMART Macro	492
BS_STREAMING Macro	492
BS_STRING_MANIPULATION_DETECTED Macro	493
BS_TCH_CDTEXT Macro	493
BS_TCH_CDTEXT_INTERNETDB Macro	493
BS_TCH_INTERNETDB Macro	493
BS_TCH_INTERNETDB_CDTEXT Macro	494
BS_TCH_NONE Macro	494
BS_TCI_ARTIST Macro	494
BS_TCI_CATEGORY Macro	494
BS_TCI_DISK_LENGTH Macro	495
BS_TCI_DISKID Macro	495
BS_TCI_EXTENDED_INFO Macro	495
BS_TCI_FRAME_OFFSET Macro	495
BS_TCI_GENRE Macro	496
BS_TCI_REVISION Macro	496
BS_TCI_SUBMITTED_VIA Macro	496
BS_TCI_TITLE Macro	496
BS_TCI_YEAR Macro	497
BS_TF_AUDIO Macro	497
BS_TF_DATA_MODE1 Macro	497
BS_TF_DATA_MODE2 Macro	497
BS_TM_DMA Macro	498
BS_TM_NOT_APPLICABLE Macro	498
BS_TM_PIO Macro	498
BS_TM_UNKNOWN Macro	498
BS_TRUE Macro	499
BS_UDF_PARTITION_PHYSICAL Macro	499
BS_UDF_PARTITION_SPARABLE Macro	499
BS_UDF_PARTITION_VIRTUAL Macro	500
BS_UDF_VERSION_102 Macro	500
BS_UDF_VERSION_150 Macro	500
BS_UDF_VERSION_200 Macro	501
BS_UDF_VERSION_201 Macro	501
BS_UDF_VERSION_250 Macro	501
BS_UDF_VERSION_260 Macro	501
BS_UDFOPT_ALL Macro	502
BS_UDFOPT_AVCHD_DISC Macro	502
BS_UDFOPT_FILE_STREAMS Macro	502

BS_UDFOPT_IMPLEMENTATION_ID Macro	502
BS_UDFOPT_PARTITION_TYPE Macro	503
BS_UDFOPT_VERSION Macro	503
BS_UNDERRUN_PROTECTION Macro	503
BS_UNKNOWN Macro	503
BS_UPC_READ Macro	504
BS_VCD_INFINITE Macro	504
BS_VCD_KEY_0 Macro	504
BS_VCD_KEY_DEFAULT Macro	504
BS_VCD_KEY_NEXT Macro	505
BS_VCD_KEY_PREVIOUS Macro	505
BS_VCD_KEY_RETURN Macro	505
BS_VCD_SEGMENT_ITEM_0 Macro	505
BS_VCD_TRACK_ITEM_0 Macro	506
BS_WHOLE_PATH Macro	506
BS_WM_DAO Macro	506
BS_WM.DAO96 Macro	506
BS_WM_TAO Macro	507
BS_WRITE_BLURAY_R Macro	507
BS_WRITE_BLURAY_R_XL Macro	507
BS_WRITE_BLURAY_RE Macro	507
BS_WRITE_BLURAY_RE_XL Macro	508
BS_WRITE_CDR Macro	508
BS_WRITE_CDRW Macro	508
BS_WRITE_CDRW_CAV Macro	508
BS_WRITE_DVD_DL Macro	509
BS_WRITE_DVD_MRDL Macro	509
BS_WRITE_DVD_RDL_PLUS Macro	509
BS_WRITE_DVD_RWDL_PLUS Macro	509
BS_WRITE_DVDR Macro	510
BS_WRITE_DVDR_PLUS Macro	510
BS_WRITE_DVDRAM Macro	510
BS_WRITE_DVDRW Macro	510
BS_WRITE_DVDRW_PLUS Macro	511
BS_WRITE_HDDVD_R Macro	511
BS_WRITE_HDDVD_RW Macro	511
BS_WRITE_MOUNT_RAINER Macro	511
BS_WRITE_TEST Macro	512
COMPRESS_BLOCKSIZE Macro	512
ENCRYPT_BLOCKSIZE Macro	512
FOpen Macro	512
INSERT_STRUCTURE_PADDING Macro	513

int2bool Macro	513
int2BS_BOOL Macro	513
PATHSEPSTRING Macro	513
SPrintf Macro	514
StrCmp Macro	514
StrCpy Macro	514
StrLen Macro	514
StrnSet Macro	515
STRUCTURE_PADDING_BIG Macro	515
STRUCTURE_PADDING_HUGE Macro	515
STRUCTURE_PADDING_NORMAL Macro	515
STRUCTURE_PADDING_SMALL Macro	516
STRUCTURE_PADDING_STRING_DATA Macro	516
VsPrintf Macro	516
<b>Files</b>	<b>516</b>
IsoSDKBurningLib.h	517
IsoSDKDefinitions.h	517
IsoSDKExport.h	577
IsoSDKUnicode.h	587

# Index

a

# 1 Symbol Reference

## 1.1 Classes

The following table lists classes in this documentation.

### Structures

	Name	Description
	SAudioGrabbingParams ( <a href="#">see page 2</a> )	IsoSDK sub class that contains information for audio grabbing
	SBootInfoEx ( <a href="#">see page 6</a> )	IsoSDK sub class is used to get and set the parameters for boot disc.
	SBootVolumeInfo ( <a href="#">see page 10</a> )	IsoSDK sub class that contains the information about the boot volume information.
	SCompressEncryptInfo ( <a href="#">see page 11</a> )	IsoSDK sub class that is used to set and get the parameters for a compressed and/or encrypted data disc.
	SCreateImageParams ( <a href="#">see page 14</a> )	IsoSDK sub class that contains information for the ImageCreate operation.
	SDirToAdd ( <a href="#">see page 16</a> )	IsoSDK sub class that contains the information about the directory you want to add to the project.
	SDirToCreate ( <a href="#">see page 20</a> )	IsoSDK sub class that contains the information about the directory you want to create inside the project.
	SDirToRemove ( <a href="#">see page 21</a> )	IsoSDK sub class that contains the information about the directory to be removed.
	SDirRename ( <a href="#">see page 23</a> )	IsoSDK sub class that contains the information about the directory to be renamed.
	SDiskCopyOptions ( <a href="#">see page 24</a> )	IsoSDK sub class that contains information for the DiskCopy operation.
	SDVDVideoOptions ( <a href="#">see page 28</a> )	IsoSDK sub class that contains additional information for VideoDVD projects.
	SExtendedDeviceInformation ( <a href="#">see page 29</a> )	IsoSDK sub class that contains the extended information for the device.
	SFileAudioProperty ( <a href="#">see page 34</a> )	IsoSDK sub class that contains information about CD-Text and track properties.
	SFileDateTime ( <a href="#">see page 38</a> )	IsoSDK sub class that contains the information about the time & date of the specific file.
	SFileEntry ( <a href="#">see page 40</a> )	IsoSDK sub class that contains the information about the file to be used.
 	SFileTypeEx ( <a href="#">see page 44</a> )	This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.  You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes ( <a href="#">see page 211</a> ) method.  This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.
	SFileToAdd ( <a href="#">see page 47</a> )	IsoSDK sub class that contains the information about the file you want to add to the project.
	SFileToAddEx ( <a href="#">see page 50</a> )	IsoSDK sub class that contains the information about the file segment you want to add to the project.
	SFileToRemove ( <a href="#">see page 53</a> )	IsoSDK sub class that contains the information about the file to be removed.

	SFileToRename ( <a href="#">see page 54</a> )	IsoSDK sub class contains the information about the file to be renamed.
	SISOInfoEx ( <a href="#">see page 56</a> )	This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files. You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes ( <a href="#">see page 211</a> ) method. This settings will not change old added files / directories only those that get added after calling this. Will get reset with define a new project.
	SISOVolumeInfo ( <a href="#">see page 64</a> )	IsoSDK sub class that contains the information about the ISO information.
	SMediumInfo ( <a href="#">see page 67</a> )	IsoSDK sub class that contains the information about the medium.
	SOPTIONS ( <a href="#">see page 75</a> )	IsoSDK sub class that is used to get and set the parameters for the current project.
	SRAWTrackToAdd ( <a href="#">see page 82</a> )	IsoSDK sub class that contains the information of a RAW image.
	SReadErrorCorrectionParams ( <a href="#">see page 86</a> )	IsoSDK sub class that contains the information of the ReadErrorCorrection for DiskCopy and CreateImage ( <a href="#">see page 127</a> ).
	SSessionInfo ( <a href="#">see page 88</a> )	IsoSDK sub class that contains the information about the selected session.
	SSpeed ( <a href="#">see page 90</a> )	IsoSDK sub class that contains information about the possible burning speeds.
	STags ( <a href="#">see page 92</a> )	IsoSDK sub class that
	STrackInfo ( <a href="#">see page 94</a> )	IsoSDK sub class that contains the information about the selected track.
	SUDFOptions ( <a href="#">see page 97</a> )	IsoSDK sub class that is used to get and set the parameters for UDF disc.
	SUDFVolumeInfo ( <a href="#">see page 100</a> )	IsoSDK sub class that contains the information about the UDF information.
	SVideoFormat ( <a href="#">see page 105</a> )	IsoSDK sub class that contains information of a mpeg video file.

## 1.1.1 SAudioGrabbingParams Structure

### C++

```
struct SAudioGrabbingParams {
    int32 nBitrate;
    int32 nMinBitrate;
    int32 nMaxBitrate;
    int32 nQuality;
    uint8 nBitrateType;
    uint8 nEncoderType;
    uint8 nTagChoice;
    int32 nNetworkTagsHandle;
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK sub class that contains information for audio grabbing

## 1.1.1.1 SAudioGrabbingParams Members

The following tables list the members exposed by SAudioGrabbingParams.

### Public Data Members

	Name	Description
◆	nBitrate ( <a href="#">see page 4</a> )	The bitrate value for the CBR encoding.
◆	nBitrateType ( <a href="#">see page 4</a> )	The type how the IsoSDK will encode the file.  BS_ET_MP3 ( <a href="#">see page 332</a> ) BS_ET_AAC ( <a href="#">see page 332</a> ) BS_ET_OGG ( <a href="#">see page 333</a> ) BS_ET_OPUS ( <a href="#">see page 333</a> ) BS_ET_FLAC ( <a href="#">see page 332</a> ) BS_ET_WMA ( <a href="#">see page 333</a> ) BS_ET_MP4 ( <a href="#">see page 332</a> )
◆	nEncoderType ( <a href="#">see page 5</a> )	Value to specify the type of encoding.  BS_BT_VARIABLE ( <a href="#">see page 313</a> ) BS_BT_CONSTANT ( <a href="#">see page 313</a> ) BS_BT_AVERAGE ( <a href="#">see page 313</a> )
◆	nMaxBitrate ( <a href="#">see page 5</a> )	The max value for the VBR encoding.
◆	nMinBitrate ( <a href="#">see page 5</a> )	The min value for the VBR encoding.
◆	nNetworkTagsHandle ( <a href="#">see page 5</a> )	Handle of internet data base tags container. See TagsFromNetworkDialog ( <a href="#">see page 229</a> ) function.
◆	nQuality ( <a href="#">see page 5</a> )	A numeric value to define the Quality for audio encoding. For most codec the value is 1-6
◆	nTagChoice ( <a href="#">see page 6</a> )	The type how the IsoSDK will receive write the tags.  BS_TCH_NONE ( <a href="#">see page 494</a> ) BS_TCH_CDTEXT ( <a href="#">see page 493</a> ) BS_TCH_INTERNETDB ( <a href="#">see page 493</a> ) BS_TCH_CDTEXT_INTERNETDB ( <a href="#">see page 493</a> ) BS_TCH_INTERNETDB_CDTEXT ( <a href="#">see page 494</a> )

### Public Methods

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 6</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SAudioGrabbingParams.
◆	SAudioGrabbingParams ( <a href="#">see page 3</a> )	This is SAudioGrabbingParams, a member of class SAudioGrabbingParams.

## 1.1.1.2 SAudioGrabbingParams::SAudioGrabbingParams Constructor

### C++

```
SAudioGrabbingParams();
```

### Description

This is SAudioGrabbingParams, a member of class SAudioGrabbingParams.

## 1.1.1.3 SAudioGrabbingParams Data Members

The data members of the SAudioGrabbingParams class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	nBitrate ( <a href="#">see page 4</a> )	The bitrate value for the CBR encoding.
◆	nBitrateType ( <a href="#">see page 4</a> )	The type how the IsoSDK will encode the file.  BS_ET_MP3 ( <a href="#">see page 332</a> ) BS_ET_AAC ( <a href="#">see page 332</a> ) BS_ET_OGG ( <a href="#">see page 333</a> ) BS_ET_OPUS ( <a href="#">see page 333</a> ) BS_ET_FLAC ( <a href="#">see page 332</a> ) BS_ET_WMA ( <a href="#">see page 333</a> ) BS_ET_MP4 ( <a href="#">see page 332</a> )
◆	nEncoderType ( <a href="#">see page 5</a> )	Value to specify the type of encoding.  BS_BT_VARIABLE ( <a href="#">see page 313</a> ) BS_BT_CONSTANT ( <a href="#">see page 313</a> ) BS_BT_AVERAGE ( <a href="#">see page 313</a> )
◆	nMaxBitrate ( <a href="#">see page 5</a> )	The max value for the VBR encoding.
◆	nMinBitrate ( <a href="#">see page 5</a> )	The min value for the VBR encoding.
◆	nNetworkTagsHandle ( <a href="#">see page 5</a> )	Handle of internet data base tags container. See TagsFromNetworkDialog ( <a href="#">see page 229</a> ) function.
◆	nQuality ( <a href="#">see page 5</a> )	A numeric value to define the Quality for audio encoding. For most codec the value is 1-6
◆	nTagChoice ( <a href="#">see page 6</a> )	The type how the IsoSDK will receive write the tags.  BS_TCH_NONE ( <a href="#">see page 494</a> ) BS_TCH_CDTEXT ( <a href="#">see page 493</a> ) BS_TCH_INTERNETDB ( <a href="#">see page 493</a> ) BS_TCH_CDTEXT_INTERNETDB ( <a href="#">see page 493</a> ) BS_TCH_INTERNETDB_CDTEXT ( <a href="#">see page 494</a> )

**1.1.1.3.1 SAudioGrabbingParams::nBitrate Data Member****C++**

```
int32 nBitrate;
```

**Description**

The bitrate value for the CBR encoding.

**1.1.1.3.2 SAudioGrabbingParams::nBitrateType Data Member****C++**

```
uint8 nBitrateType;
```

**Description**

The type how the IsoSDK will encode the file.

BS\_ET\_MP3 ([see page 332](#))

BS\_ET\_AAC ([see page 332](#))

BS\_ET\_OGG ([see page 333](#))

BS\_ET\_OPUS ([see page 333](#))

BS\_ET\_FLAC ( [see page 332](#))  
BS\_ET\_WMA ( [see page 333](#))  
BS\_ET\_MP4 ( [see page 332](#))

### 1.1.1.3.3 SAudioGrabbingParams::nEncoderType Data Member

#### C++

```
uint8 nEncoderType;
```

#### Description

Value to specify the type of encoding.

BS\_BT\_VARIABLE ( [see page 313](#))  
BS\_BT\_CONSTANT ( [see page 313](#))  
BS\_BT\_AVERAGE ( [see page 313](#))

### 1.1.1.3.4 SAudioGrabbingParams::nMaxBitrate Data Member

#### C++

```
int32 nMaxBitrate;
```

#### Description

The max value for the VBR encoding.

### 1.1.1.3.5 SAudioGrabbingParams::nMinBitrate Data Member

#### C++

```
int32 nMinBitrate;
```

#### Description

The max value for the VBR encoding.

### 1.1.1.3.6 SAudioGrabbingParams::nNetworkTagsHandle Data Member

#### C++

```
int32 nNetworkTagsHandle;
```

#### Description

Handle of internet data base tags container. See TagsFromNetworkDialog ( [see page 229](#)) function.

### 1.1.1.3.7 SAudioGrabbingParams::nQuality Data Member

#### C++

```
int32 nQuality;
```

#### Description

A numeric value to define the Quality for audio encoding. For most codec the value is 1-6

### 1.1.1.3.8 SAudioGrabbingParams::nTagChoice Data Member

**C++**

```
uint8 nTagChoice;
```

#### Description

The type how the IsoSDK will receive write the tags.

BS\_TCH\_NONE ([see page 494](#))

BS\_TCH\_CDTEXT ([see page 493](#))

BS\_TCH\_INTERNETDB ([see page 493](#))

BS\_TCH\_CDTEXT\_INTERNETDB ([see page 493](#))

BS\_TCH\_INTERNETDB\_CDTEXT ([see page 494](#))

### 1.1.1.4 SAudioGrabbingParams Methods

The methods of the SAudioGrabbingParams class are listed here.

#### Public Methods

	Name	Description
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 6</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SAudioGrabbingParams.

#### 1.1.1.4.1 SAudioGrabbingParams::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_HUGE);
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SAudioGrabbingParams.

---

## 1.1.2 SBootInfoEx Structure

**C++**

```
struct SBootInfoEx {
    TCHAR DeveloperID[24];
    BS_BOOL BootIndicator;
    int32 LoadSegment;
    int32 PlatformID;
    int32 Emulation;
    int32 SectorCount;
};
```

#### File

IsoSDKDefinitions.h ([see page 517](#))

#### Description

IsoSDK sub class is used to get and set the parameters for boot disc.

## 1.1.2.1 SBootInfoEx Members

The following tables list the members exposed by SBootInfoEx.

### Public Data Members

	Name	Description
◆	BootIndicator ( <a href="#">see page 8</a> )	This is byte 0 of the boot catalog. BS_TRUE ( <a href="#">see page 499</a> ) = bootable (0x88) and BS_FALSE ( <a href="#">see page 335</a> ) = not bootable (0x00).
◆	DeveloperID ( <a href="#">see page 8</a> )	These are the bytes 4-1F of the header entry. It contains the string to the developer. E. g. your company name.
◆	Emulation ( <a href="#">see page 8</a> )	This is byte 1 from the boot catalog. It sets the boot media type. It specifies what media the boot image is intended to emulate. 0 = No emulation 1 = 1.2 MB diskette 2 = 1.44 MB diskette 3 = 2.88 MB diskette 4 = hard disk
◆	LoadSegment ( <a href="#">see page 9</a> )	These are the bytes 2-3 of the boot catalog. The default value is 1984 (7C0). It is the address of the load segment for the initial boot image.
◆	PlatformID ( <a href="#">see page 9</a> )	The platform ID, byte 1 of the header entry: 0 = 80x86 1 = Power PC 2 = Mac
◆	SectorCount ( <a href="#">see page 9</a> )	These are the bytes 6-7 of the boot catalog. They represent the number of virtual/emulated sectors the system will store at the load segment.

### Public Methods

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 9</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SBootInfoEx.
◆	SBootInfoEx ( <a href="#">see page 7</a> )	This is SBootInfoEx, a member of class SBootInfoEx.

## 1.1.2.2 SBootInfoEx::SBootInfoEx Constructor

### C++

```
SBootInfoEx();
```

### Description

This is SBootInfoEx, a member of class SBootInfoEx.

## 1.1.2.3 SBootInfoEx Data Members

The data members of the SBootInfoEx class are listed here.

### Public Data Members

	Name	Description
◆	BootIndicator ( <a href="#">see page 8</a> )	This is byte 0 of the boot catalog. BS_TRUE ( <a href="#">see page 499</a> ) = bootable (0x88) and BS_FALSE ( <a href="#">see page 335</a> ) = not bootable (0x00).
◆	DeveloperID ( <a href="#">see page 8</a> )	These are the bytes 4-1F of the header entry. It contains the string to the developer. E. g. your company name.

◆	Emulation (see page 8)	This is byte 1 from the boot catalog. It sets the boot media type. It specifies what media the boot image is intended to emulate.  0 = No emulation 1 = 1.2 MB diskette 2 = 1.44 MB diskette 3 = 2.88 MB diskette 4 = hard disk
◆	LoadSegment (see page 9)	These are the bytes 2-3 of the boot catalog. The default value is 1984 (7C0). It is the address of the load segment for the initial boot image.
◆	PlatformID (see page 9)	The platform ID, byte 1 of the header entry:  0 = 80x86 1 = Power PC 2 = Mac
◆	SectorCount (see page 9)	These are the bytes 6-7 of the boot catalog. They represent the number of virtual/emulated sectors the system will store at the load segment.

### 1.1.2.3.1 SBootInfoEx::BootIndicator Data Member

C++

```
BS_BOOL BootIndicator;
```

**Description**

This is byte 0 of the boot catalog. BS\_TRUE (see page 499) = bootable (0x88) and BS\_FALSE (see page 335) = not bootable (0x00).

### 1.1.2.3.2 SBootInfoEx::DeveloperID Data Member

C++

```
TCHAR DeveloperID[ 24 ];
```

**Description**

These are the bytes 4-1F of the header entry. It contains the string to the developer. E. g. your company name.

### 1.1.2.3.3 SBootInfoEx::Emulation Data Member

C++

```
int32 Emulation;
```

**Description**

This is byte 1 from the boot catalog. It sets the boot media type. It specifies what media the boot image is intended to emulate.

0 = No emulation

1 = 1.2 MB diskette

2 = 1.44 MB diskette

3 = 2.88 MB diskette

4 = hard disk

### 1.1.2.3.4 SBootInfoEx::LoadSegment Data Member

C++

```
int32 LoadSegment;
```

**Description**

These are the bytes 2-3 of the boot catalog. The default value is 1984 (7C0). It is the address of the load segment for the initial boot image.

### 1.1.2.3.5 SBootInfoEx::PlatformID Data Member

C++

```
int32 PlatformID;
```

**Description**

The platform ID, byte 1 of the header entry:

0 = 80x86

1 = Power PC

2 = Mac

### 1.1.2.3.6 SBootInfoEx::SectorCount Data Member

C++

```
int32 SectorCount;
```

**Description**

These are the bytes 6-7 of the boot catalog. They represent the number of virtual/emulated sectors the system will store at the load segment.

## 1.1.2.4 SBootInfoEx Methods

The methods of the SBootInfoEx class are listed here.

**Public Methods**

	Name	Description
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 9</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SBootInfoEx.

### 1.1.2.4.1 SBootInfoEx::INSERT\_STRUCTURE\_PADDING Method

C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_HUGE);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SBootInfoEx.

## 1.1.3 SBootVolumeInfo Structure

### C++

```
struct SBootVolumeInfo {
    int32 nVolumeDescriptorAddress;
    struct SBootInfoEx sInfoEx;
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK sub class that contains the information about the boot volume information.

### 1.1.3.1 SBootVolumeInfo Members

The following tables list the members exposed by SBootVolumeInfo.

#### Public Data Members

	Name	Description
◆	nVolumeDescriptorAddress ( <a href="#">see page 11</a> )	The address where this descriptor starts.
◆	sInfoEx ( <a href="#">see page 11</a> )	A structure SBootInfoEx ( <a href="#">see page 6</a> ).

#### Public Methods

	Name	Description
≡◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 11</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SBootVolumeInfo.
≡◆	SBootVolumeInfo ( <a href="#">see page 10</a> )	This is SBootVolumeInfo, a member of class SBootVolumeInfo.

### 1.1.3.2 SBootVolumeInfo::SBootVolumeInfo Constructor

### C++

```
SBootVolumeInfo();
```

### Description

This is SBootVolumeInfo, a member of class SBootVolumeInfo.

### 1.1.3.3 SBootVolumeInfo Data Members

The data members of the SBootVolumeInfo class are listed here.

#### Public Data Members

	Name	Description
◆	nVolumeDescriptorAddress ( <a href="#">see page 11</a> )	The address where this descriptor starts.
◆	sInfoEx ( <a href="#">see page 11</a> )	A structure SBootInfoEx ( <a href="#">see page 6</a> ).

### 1.1.3.3.1 SBootVolumeInfo::nVolumeDescriptorAddress Data Member

**C++**

```
int32 nVolumeDescriptorAddress;
```

**Description**

The address where this descriptor starts.

### 1.1.3.3.2 SBootVolumeInfo::sInfoEx Data Member

**C++**

```
struct SBootInfoEx sInfoEx;
```

**Description**

A structure SBootInfoEx (see page 6).

## 1.1.3.4 SBootVolumeInfo Methods

The methods of the SBootVolumeInfo class are listed here.

**Public Methods**

	Name	Description
≡	INSERT_STRUCTURE_PADDING ( see page 11)	This is INSERT_STRUCTURE_PADDING, a member of class SBootVolumeInfo.

### 1.1.3.4.1 SBootVolumeInfo::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_HUGE);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SBootVolumeInfo.

---

## 1.1.4 SCompressEncryptInfo Structure

**C++**

```
struct SCompressEncryptInfo {
    BS_BOOL bCompression;
    int32 nCompressionLevel;
    BS_BOOL bEncryption;
    const char* lpszPassword;
};
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK sub class that is used to set and get the parameters for a compressed and/or encrypted data disc.

## 1.1.4.1 SCompressEncryptInfo Members

The following tables list the members exposed by SCompressEncryptInfo.

### Public Data Members

	Name	Description
◆	bCompression ( [ see page 13 )	This parameter will enable/disable the compression function. You can enable it with BS_TRUE ( [ see page 499 ) or disable it with BS_FALSE ( [ see page 335 ).
◆	bEncryption ( [ see page 13 )	This parameter will enable/disable the encryption function. You can enable it with BS_TRUE ( [ see page 499 ) or disable it with BS_FALSE ( [ see page 335 ).
◆	lpszPassword ( [ see page 13 )	This is the password string to decrypt the encryption later. This value is only valid upon setting. If you call to get this string it is empty due to security issues.
◆	nCompressionLevel ( [ see page 13 )	The range of the compression level. The compression level range is 0 up to 9. 0 is fast compression and 9 is slow but more effective compression. Invalid values are changed to the nearest values to avoid damaged disc.

### Public Methods

	Name	Description
≡	INSERT_STRUCTURE_PADDING ( [ see page 13 )	This is INSERT_STRUCTURE_PADDING, a member of class SCompressEncryptInfo.
≡	SCompressEncryptInfo ( [ see page 12 )	This is SCompressEncryptInfo, a member of class SCompressEncryptInfo.

## 1.1.4.2 SCompressEncryptInfo::SCompressEncryptInfo Constructor

### C++

```
SCompressEncryptInfo();
```

### Description

This is SCompressEncryptInfo, a member of class SCompressEncryptInfo.

## 1.1.4.3 SCompressEncryptInfo Data Members

The data members of the SCompressEncryptInfo class are listed here.

### Public Data Members

	Name	Description
◆	bCompression ( [ see page 13 )	This parameter will enable/disable the compression function. You can enable it with BS_TRUE ( [ see page 499 ) or disable it with BS_FALSE ( [ see page 335 ).
◆	bEncryption ( [ see page 13 )	This parameter will enable/disable the encryption function. You can enable it with BS_TRUE ( [ see page 499 ) or disable it with BS_FALSE ( [ see page 335 ).
◆	lpszPassword ( [ see page 13 )	This is the password string to decrypt the encryption later. This value is only valid upon setting. If you call to get this string it is empty due to security issues.
◆	nCompressionLevel ( [ see page 13 )	The range of the compression level. The compression level range is 0 up to 9. 0 is fast compression and 9 is slow but more effective compression. Invalid values are changed to the nearest values to avoid damaged disc.

### 1.1.4.3.1 SCompressEncryptInfo::bCompression Data Member

**C++**

```
BS_BOOL bCompression;
```

**Description**

This parameter will enable/disable the compression function. You can enable it with BS\_TRUE (↗ see page 499) or disable it with BS\_FALSE (↗ see page 335).

### 1.1.4.3.2 SCompressEncryptInfo::bEncryption Data Member

**C++**

```
BS_BOOL bEncryption;
```

**Description**

This parameter will enable/disable the encryption function. You can enable it with BS\_TRUE (↗ see page 499) or disable it with BS\_FALSE (↗ see page 335).

### 1.1.4.3.3 SCompressEncryptInfo::lpszPassword Data Member

**C++**

```
const char* lpszPassword;
```

**Description**

This is the password string to decrypt the encryption later. This value is only valid upon setting. If you call to get this string it is empty due to security issues.

### 1.1.4.3.4 SCompressEncryptInfo::nCompressionLevel Data Member

**C++**

```
int32 nCompressionLevel;
```

**Description**

The range of the compression level. The compression level range is 0 up to 9. 0 is fast compression and 9 is slow but more effective compression. Invalid values are changed to the nearest values to avoid damaged disc.

## 1.1.4.4 SCompressEncryptInfo Methods

The methods of the SCompressEncryptInfo class are listed here.

**Public Methods**

	Name	Description
☞	INSERT_STRUCTURE_PADDING (↗ see page 13)	This is INSERT_STRUCTURE_PADDING, a member of class SCompressEncryptInfo.

### 1.1.4.4.1 SCompressEncryptInfo::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SCompressEncryptInfo.

## 1.1.5 SCreateImageParams Structure

### C++

```
struct SCreateImageParams {
    TCHAR lpszImagePath[_MAX_PATH];
    TCHAR lpszBadSectorsFilePath[_MAX_PATH];
    int16 nImageType;
    int32 nVerifyBufferSectors;
    BS_BOOL bFullCapacity;
    SReadErrorCorrectionParams cErrorParams;
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK sub class that contains information for the ImageCreate operation.

### 1.1.5.1 SCreateImageParams Members

The following tables list the members exposed by SCreateImageParams.

#### Public Data Members

	Name	Description
◆	bFullCapacity ( <a href="#">see page 15</a> )	Set to true to read the full disc. When set to false, the IsoSDK will read the track size. Audio, Mixed Mode and VideoCD always set to false
◆	cErrorParams ( <a href="#">see page 15</a> )	A structure SReadErrorCorrectionParams ( <a href="#">see page 86</a> )
◆	lpszBadSectorsFilePath ( <a href="#">see page 15</a> )	If you fill this parameter with a path and filename, the IsoSDK will create a textfile with a list of BadSectors.
◆	lpszImagePath ( <a href="#">see page 15</a> )	The path and filename of the target disc image.
◆	nImageType ( <a href="#">see page 16</a> )	The type of disc image. Possible values are: BS_IMG_ISO ( <a href="#">see page 358</a> ), BS_IMG_BIN ( <a href="#">see page 358</a> )
◆	nVerifyBufferSectors ( <a href="#">see page 16</a> )	This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.

#### Public Methods

	Name	Description
≡◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 16</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SCreateImageParams.
≡◆	SCreateImageParams ( <a href="#">see page 14</a> )	This is SCreateImageParams, a member of class SCreateImageParams.

### 1.1.5.2 SCreateImageParams::SCreateImageParams Constructor

### C++

```
SCreateImageParams();
```

### Description

This is SCreateImageParams, a member of class SCreateImageParams.

## 1.1.5.3 SCreateImageParams Data Members

The data members of the SCreateImageParams class are listed here.

### Public Data Members

	Name	Description
◆	bFullCapacity ( <a href="#"> see page 15</a> )	Set to true to read the full disc. When set to false, the IsoSDK will read the track size. Audio, Mixed Mode and VideoCD always set to false
◆	cErrorParams ( <a href="#"> see page 15</a> )	A structure SReadErrorCorrectionParams ( <a href="#"> see page 86</a> )
◆	lpszBadSectorsFilePath ( <a href="#"> see page 15</a> )	If you fill this parameter with a path and filename, the IsoSDK will create a textfile with a list of BadSectors.
◆	lpszImagePath ( <a href="#"> see page 15</a> )	The path and filename of the target disc image.
◆	nImageType ( <a href="#"> see page 16</a> )	The type of disc image. Possible values are: BS_IMG_ISO ( <a href="#"> see page 358</a> ), BS_IMG_BIN ( <a href="#"> see page 358</a> )
◆	nVerifyBufferSectors ( <a href="#"> see page 16</a> )	This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.

### 1.1.5.3.1 SCreateImageParams::bFullCapacity Data Member

#### C++

```
BS_BOOL bFullCapacity;
```

#### Description

Set to true to read the full disc. When set to false, the IsoSDK will read the track size.

Audio, Mixed Mode and VideoCD always set to false

### 1.1.5.3.2 SCreateImageParams::cErrorParams Data Member

#### C++

```
SReadErrorCorrectionParams cErrorParams;
```

#### Description

A structure SReadErrorCorrectionParams ( [see page 86](#))

### 1.1.5.3.3 SCreateImageParams::lpszBadSectorsFilePath Data Member

#### C++

```
TCHAR lpszBadSectorsFilePath[_MAX_PATH];
```

#### Description

If you fill this parameter with a path and filename, the IsoSDK will create a textfile with a list of BadSectors.

### 1.1.5.3.4 SCreateImageParams::lpszImagePath Data Member

#### C++

```
TCHAR lpszImagePath[_MAX_PATH];
```

#### Description

The path and filename of the target disc image.

### 1.1.5.3.5 SCreatelImageParams::nImageType Data Member

C++

```
int16 nImageType;
```

**Description**

The type of disc image. Possible values are: BS\_IMG\_ISO (see page 358), BS\_IMG\_BIN (see page 358)

### 1.1.5.3.6 SCreatelImageParams::nVerifyBufferSectors Data Member

C++

```
int32 nVerifyBufferSectors;
```

**Description**

This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.

## 1.1.5.4 SCreatelImageParams Methods

The methods of the SCreatelImageParams class are listed here.

**Public Methods**

	Name	Description
☞	INSERT_STRUCTURE_PADDING (see page 16)	This is INSERT_STRUCTURE_PADDING, a member of class SCreatelImageParams.

### 1.1.5.4.1 SCreatelImageParams::INSERT\_STRUCTURE\_PADDING Method

C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - sizeof(int32) - sizeof(BS_BOOL));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SCreatelImageParams.

---

## 1.1.6 SDirToAdd Structure

C++

```
struct SDirToAdd {
    const TCHAR* lpszSourceDirPath;
    const TCHAR* lpszDestinationPath;
    const TCHAR* lpszFileSpecification;
    int32 nFileAttributes;
    BS_BOOL bRecursive;
    int32 nSavePath;
};
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK sub class that contains the information about the directory you want to add to the project.

## 1.1.6.1 SDirToAdd Members

The following tables list the members exposed by SDirToAdd.

### Public Data Members

	Name	Description
◆	bRecursive ( <a href="#">see page 18</a> )	This argument states whether the IsoSDK will add the underlaying file. Recursive = BS_TRUE ( <a href="#">see page 499</a> ) (default). If you only want to add the named folder you have to set it to BS_FALSE ( <a href="#">see page 335</a> ).
◆	lpszDestinationPath ( <a href="#">see page 18</a> )	Path of the destination folder of the project. (null terminated).
◆	lpszFileSpecification ( <a href="#">see page 18</a> )	Specifies the file filter to be applied to the folder. Examples: „*.*“ or „*.doc“ or „MyPrefix_*.*“ (null terminated).
◆	lpszSourceDirPath ( <a href="#">see page 18</a> )	The folder path you want to add to the project.
◆	nFileAttributes ( <a href="#">see page 19</a> )	You can also set file attributes to filter files in the specified folder: BS_FA_READONLY ( <a href="#">see page 335</a> ), BS_FA_HIDDEN ( <a href="#">see page 334</a> ), BS_FA_ADVANCED_HIDDEN ( <a href="#">see page 333</a> ), BS_FA_SYSTEM ( <a href="#">see page 335</a> ), BS_FA_DIRECTORY ( <a href="#">see page 334</a> ), BS_FA_ARCHIVE ( <a href="#">see page 334</a> ), BS_FA_ALL ( <a href="#">see page 334</a> )
◆	nSavePath ( <a href="#">see page 19</a> )	This argument states if and how the super ordinate directory is added as a folder to the project: BS_DONT_SAVE_PATH ( <a href="#">see page 326</a> ), BS_WHOLE_PATH ( <a href="#">see page 506</a> ), BS_PARENTDIR_ONLY ( <a href="#">see page 367</a> )

### Public Methods

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 19</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToAdd.
◆	SDirToAdd ( <a href="#">see page 17</a> )	This is SDirToAdd, a member of class SDirToAdd.

## 1.1.6.2 SDirToAdd::SDirToAdd Constructor

### C++

```
SDirToAdd();
```

### Description

This is SDirToAdd, a member of class SDirToAdd.

## 1.1.6.3 SDirToAdd Data Members

The data members of the SDirToAdd class are listed here.

### Public Data Members

	Name	Description
◆	bRecursive ( <a href="#">see page 18</a> )	This argument states whether the IsoSDK will add the underlaying file. Recursive = BS_TRUE ( <a href="#">see page 499</a> ) (default). If you only want to add the named folder you have to set it to BS_FALSE ( <a href="#">see page 335</a> ).

❖	lpszDestinationPath ( <a href="#">see page 18</a> )	Path of the destination folder of the project. (null terminated).
❖	lpszFileSpecification ( <a href="#">see page 18</a> )	Specifies the file filter to be applied to the folder. Examples: <code>*.*</code> or <code>*.doc</code> or <code>„MyPrefix_*.*</code> (null terminated).
❖	lpszSourceDirPath ( <a href="#">see page 18</a> )	The folder path you want to add to the project.
❖	nFileAttributes ( <a href="#">see page 19</a> )	You can also set file attributes to filter files in the specified folder: <code>BS_FA_READONLY</code> ( <a href="#">see page 335</a> ), <code>BS_FA_HIDDEN</code> ( <a href="#">see page 334</a> ), <code>BS_FA_ADVANCED_HIDDEN</code> ( <a href="#">see page 333</a> ), <code>BS_FA_SYSTEM</code> ( <a href="#">see page 335</a> ), <code>BS_FA_DIRECTORY</code> ( <a href="#">see page 334</a> ), <code>BS_FA_ARCHIVE</code> ( <a href="#">see page 334</a> ), <code>BS_FA_ALL</code> ( <a href="#">see page 334</a> )
❖	nSavePath ( <a href="#">see page 19</a> )	This argument states if and how the super ordinate directory is added as a folder to the project: <code>BS_DONT_SAVE_PATH</code> ( <a href="#">see page 326</a> ), <code>BS_WHOLE_PATH</code> ( <a href="#">see page 506</a> ), <code>BS_PARENTDIR_ONLY</code> ( <a href="#">see page 367</a> )

### 1.1.6.3.1 SDirToAdd::bRecursive Data Member

C++

```
BS_BOOL bRecursive;
```

#### Description

This argument states whether the IsoSDK will add the underlaying file. Recursive = BS\_TRUE ([see page 499](#)) (default). If you only want to add the named folder you have to set it to BS\_FALSE ([see page 335](#)).

### 1.1.6.3.2 SDirToAdd::lpszDestinationPath Data Member

C++

```
const TCHAR* lpszDestinationPath;
```

#### Description

Path of the destination folder of the project. (null terminated).

### 1.1.6.3.3 SDirToAdd::lpszFileSpecification Data Member

C++

```
const TCHAR* lpszFileSpecification;
```

#### Description

Specifies the file filter to be applied to the folder. Examples: `*.*` or `*.doc` or `„MyPrefix_*.*` (null terminated).

### 1.1.6.3.4 SDirToAdd::lpszSourceDirPath Data Member

C++

```
const TCHAR* lpszSourceDirPath;
```

#### Description

The folder path you want to add to the project.

### 1.1.6.3.5 SDirToAdd::nFileAttributes Data Member

**C++**

```
int32 nFileAttributes;
```

**Description**

You can also set file attributes to filter files in the specified folder:

- BS\_FA\_READONLY (see page 335),
- BS\_FA\_HIDDEN (see page 334),
- BS\_FA\_ADVANCED\_HIDDEN (see page 333),
- BS\_FA\_SYSTEM (see page 335),
- BS\_FA\_DIRECTORY (see page 334),
- BS\_FA\_ARCHIVE (see page 334),
- BS\_FA\_ALL (see page 334)

### 1.1.6.3.6 SDirToAdd::nSavePath Data Member

**C++**

```
int32 nSavePath;
```

**Description**

This argument states if and how the super ordinate directory is added as a folder to the project:

- BS\_DONT\_SAVE\_PATH (see page 326),
- BS\_WHOLE\_PATH (see page 506),
- BS\_PARENTDIR\_ONLY (see page 367)

## 1.1.6.4 SDirToAdd Methods

The methods of the SDirToAdd class are listed here.

**Public Methods**

	Name	Description
💡	INSERT_STRUCTURE_PADDING (see page 19)	This is INSERT_STRUCTURE_PADDING, a member of class SDirToAdd.

### 1.1.6.4.1 SDirToAdd::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SDirToAdd.

## 1.1.7 SDirToCreate Structure

### C++

```
struct SDirToCreate {
    const TCHAR* lpszDir;
    const TCHAR* lpszDestinationPath;
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK sub class that contains the information about the directory you want to create inside the project.

### 1.1.7.1 SDirToCreate Members

The following tables list the members exposed by SDirToCreate.

#### Public Data Members

	Name	Description
❖	IpszDestinationPath ( <a href="#">see page 21</a> )	This is the destination path of your project in which you want to create the new directory.
❖	IpszDir ( <a href="#">see page 21</a> )	This is the name of the new directory. Only one plane is allowed. (e. g. „new“, but not „new\new“).

#### Public Methods

	Name	Description
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 21</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToCreate.
❖	SDirToCreate ( <a href="#">see page 20</a> )	This is SDirToCreate, a member of class SDirToCreate.

### 1.1.7.2 SDirToCreate::SDirToCreate Constructor

### C++

```
SDirToCreate();
```

### Description

This is SDirToCreate, a member of class SDirToCreate.

### 1.1.7.3 SDirToCreate Data Members

The data members of the SDirToCreate class are listed here.

#### Public Data Members

	Name	Description
❖	IpszDestinationPath ( <a href="#">see page 21</a> )	This is the destination path of your project in which you want to create the new directory.
❖	IpszDir ( <a href="#">see page 21</a> )	This is the name of the new directory. Only one plane is allowed. (e. g. „new“, but not „new\new“).

### 1.1.7.3.1 SDirToCreate::lpszDestinationPath Data Member

**C++**

```
const TCHAR* lpszDestinationPath;
```

**Description**

This is the destination path of your project in which you want to create the new directory.

### 1.1.7.3.2 SDirToCreate::lpszDir Data Member

**C++**

```
const TCHAR* lpszDir;
```

**Description**

This is the name of the new directory. Only one plane is allowed. (e. g. „new“, but not „new\new“).

## 1.1.7.4 SDirToCreate Methods

The methods of the SDirToCreate class are listed here.

**Public Methods**

	Name	Description
≡	INSERT_STRUCTURE_PADDING ( <a href="#">see page 21</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToCreate.

### 1.1.7.4.1 SDirToCreate::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SDirToCreate.

---

## 1.1.8 SDirToRemove Structure

**C++**

```
struct SDirToRemove {
    const TCHAR* lpszDestinationPath;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the directory to be removed.

### 1.1.8.1 SDirToRemove Members

The following tables list the members exposed by SDirToRemove.

**Public Data Members**

	Name	Description
◆	IpszDestinationPath ( <a href="#">see page 22</a> )	The path of the directory that you want to remove. (null terminated).

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 22</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToRemove.
◆	SDirToRemove ( <a href="#">see page 22</a> )	This is SDirToRemove, a member of class SDirToRemove.

## 1.1.8.2 SDirToRemove::SDirToRemove Constructor

**C++**

```
SDirToRemove();
```

**Description**

This is SDirToRemove, a member of class SDirToRemove.

## 1.1.8.3 SDirToRemove Data Members

The data members of the SDirToRemove class are listed here.

**Public Data Members**

	Name	Description
◆	IpszDestinationPath ( <a href="#">see page 22</a> )	The path of the directory that you want to remove. (null terminated).

### 1.1.8.3.1 SDirToRemove::IpszDestinationPath Data Member

**C++**

```
const TCHAR* lpszDestinationPath;
```

**Description**

The path of the directory that you want to remove. (null terminated).

## 1.1.8.4 SDirToRemove Methods

The methods of the SDirToRemove class are listed here.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 22</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToRemove.

### 1.1.8.4.1 SDirToRemove::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SDirToRemove.

## 1.1.9 SDirToRename Structure

**C++**

```
struct SDirToRename {
    const TCHAR* lpszSourcePath;
    const TCHAR* lpszDestinationPath;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the directory to be renamed.

### 1.1.9.1 SDirToRename Members

The following tables list the members exposed by SDirToRename.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
❖	IpszDestinationPath ( <a href="#">see page 24</a> )	Pointer to the null-terminated string with the destination path of the directory you want to rename/move.
❖	IpszSourcePath ( <a href="#">see page 24</a> )	Pointer to the null-terminated string with the source path of the directory you want to rename/move.

**Public Methods**

	<b>Name</b>	<b>Description</b>
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 24</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDirToRename.
❖	SDirToRename ( <a href="#">see page 23</a> )	This is SDirToRename, a member of class SDirToRename.

### 1.1.9.2 SDirToRename::SDirToRename Constructor

**C++**

```
SDirToRename();
```

**Description**

This is SDirToRename, a member of class SDirToRename.

### 1.1.9.3 SDirToRename Data Members

The data members of the SDirToRename class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
❖	IpszDestinationPath ( <a href="#">see page 24</a> )	Pointer to the null-terminated string with the destination path of the directory you want to rename/move.

	<code>lpszSourcePath</code> ( <a href="#">see page 24</a> )	Pointer to the null-terminated string with the source path of the directory you want to rename/move.
---	---	--

### 1.1.9.3.1 SDirToRename::lpszDestinationPath Data Member

C++

```
const TCHAR* lpszDestinationPath;
```

**Description**

Pointer to the null-terminated string with the destination path of the directory you want to rename/move.

### 1.1.9.3.2 SDirToRename::lpszSourcePath Data Member

C++

```
const TCHAR* lpszSourcePath;
```

**Description**

Pointer to the null-terminated string with the source path of the directory you want to rename/move.

## 1.1.9.4 SDirToRename Methods

The methods of the SDirToRename class are listed here.

**Public Methods**

	Name	Description
	<code>INSERT_STRUCTURE_PADDING</code> ( <a href="#">see page 24</a> )	This is <code>INSERT_STRUCTURE_PADDING</code> , a member of class SDirToRename.

### 1.1.9.4.1 SDirToRename::INSERT\_STRUCTURE\_PADDING Method

C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is `INSERT_STRUCTURE_PADDING`, a member of class SDirToRename.

---

## 1.1.10 SDiskCopyOptions Structure

C++

```
struct SDiskCopyOptions {
    uint8 nWriteMethod;
    uint8 nReadMode;
    BS_BOOL bVerifyAfterBurn;
    SReadErrorCorrectionParams cErrorParams;
    BS_BOOL bEjectAfterBurn;
    int32 nVerifyBufferSectors;
    BS_BOOL bFullCapacity;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains information for the DiskCopy operation.

## 1.1.10.1 SDiskCopyOptions Members

The following tables list the members exposed by SDiskCopyOptions.

**Public Data Members**

	Name	Description
◆	bEjectAfterBurn ( <a href="#">see page 26</a> )	This argument determines whether the medium will be ejected after the burning process. BS_TRUE ( <a href="#">see page 499</a> ) ejects the medium, and BS_FALSE ( <a href="#">see page 335</a> ) prevents this. This is special for multi duplicator systems to disable the automatic eject there.
◆	bFullCapacity ( <a href="#">see page 26</a> )	Set to true to read the full disc. When set to false, the SDK will read the track size. Audio, Mixed Mode and VideoCD always set to false.
◆	bVerifyAfterBurn ( <a href="#">see page 26</a> )	Indicates to verify (TRUE) the target disc after burning.
◆	cErrorParams ( <a href="#">see page 27</a> )	A structure SReadErrorCorrectionParams ( <a href="#">see page 86</a> ).
◆	nReadMode ( <a href="#">see page 27</a> )	The read mode of the source disc. Possible values are: BS_RM_USERDATA ( <a href="#">see page 377</a> ), BS_RM_RAW ( <a href="#">see page 376</a> ), BS_RM_RAW_SUBCHANNEL ( <a href="#">see page 376</a> )
◆	nVerifyBufferSectors ( <a href="#">see page 27</a> )	This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.
◆	nWriteMethod ( <a href="#">see page 27</a> )	The write method for the target device. Allowed are BS_WM_DAO ( <a href="#">see page 506</a> ) and BS_WM_DAO96 ( <a href="#">see page 506</a> ) where BS_WM_DAO96 ( <a href="#">see page 506</a> ) is only used if the ReadMode is BS_RM_RAW_SUBCHANNEL ( <a href="#">see page 376</a> ).

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 27</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDiskCopyOptions.
◆	SDiskCopyOptions ( <a href="#">see page 25</a> )	This is SDiskCopyOptions, a member of class SDiskCopyOptions.

## 1.1.10.2 SDiskCopyOptions::SDiskCopyOptions Constructor

**C++**

```
SDiskCopyOptions();
```

**Description**

This is SDiskCopyOptions, a member of class SDiskCopyOptions.

## 1.1.10.3 SDiskCopyOptions Data Members

The data members of the SDiskCopyOptions class are listed here.

**Public Data Members**

<b>Name</b>	<b>Description</b>
◆ bEjectAfterBurn ( <a href="#">see page 26</a> )	This argument determines whether the medium will be ejected after the burning process. BS_TRUE ( <a href="#">see page 499</a> ) ejects the medium, and BS_FALSE ( <a href="#">see page 335</a> ) prevents this. This is special for multi duplicator systems to disable the automatic eject there.
◆ bFullCapacity ( <a href="#">see page 26</a> )	Set to true to read the full disc. When set to false, the SDK will read the track size. Audio, Mixed Mode and VideoCD always set to false.
◆ bVerifyAfterBurn ( <a href="#">see page 26</a> )	Indicates to verify (TRUE) the target disc after burning.
◆ cErrorParams ( <a href="#">see page 27</a> )	A structure SReadErrorCorrectionParams ( <a href="#">see page 86</a> ).
◆ nReadMode ( <a href="#">see page 27</a> )	The read mode of the source disc. Possible values are: BS_RM_USERDATA ( <a href="#">see page 377</a> ), BS_RM_RAW ( <a href="#">see page 376</a> ), BS_RM_RAW_SUBCHANNEL ( <a href="#">see page 376</a> )
◆ nVerifyBufferSectors ( <a href="#">see page 27</a> )	This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.
◆ nWriteMethod ( <a href="#">see page 27</a> )	The write method for the target device. Allowed are BS_WM_DAO ( <a href="#">see page 506</a> ) and BS_WM.DAO96 ( <a href="#">see page 506</a> ) where BS_WM.DAO96 ( <a href="#">see page 506</a> ) is only used if the ReadMode is BS_RM_RAW_SUBCHANNEL ( <a href="#">see page 376</a> ).

**1.1.10.3.1 SDiskCopyOptions::bEjectAfterBurn Data Member****C++**

```
BS_BOOL bEjectAfterBurn;
```

**Description**

This argument determines whether the medium will be ejected after the burning process. BS\_TRUE ([see page 499](#)) ejects the medium, and BS\_FALSE ([see page 335](#)) prevents this.

This is special for multi duplicator systems to disable the automatic eject there.

**1.1.10.3.2 SDiskCopyOptions::bFullCapacity Data Member****C++**

```
BS_BOOL bFullCapacity;
```

**Description**

Set to true to read the full disc. When set to false, the SDK will read the track size.

Audio, Mixed Mode and VideoCD always set to false.

**1.1.10.3.3 SDiskCopyOptions::bVerifyAfterBurn Data Member****C++**

```
BS_BOOL bVerifyAfterBurn;
```

**Description**

Indicates to verify (TRUE) the target disc after burning.

### 1.1.10.3.4 SDiskCopyOptions::cErrorParams Data Member

**C++**

```
SReadErrorCorrectionParams cErrorParams;
```

**Description**

A structure SReadErrorCorrectionParams ([see page 86](#)).

### 1.1.10.3.5 SDiskCopyOptions::nReadMode Data Member

**C++**

```
uint8 nReadMode;
```

**Description**

The read mode of the source disc. Possible values are:

BS\_RM\_USERDATA ([see page 377](#)),

BS\_RM\_RAW ([see page 376](#)),

BS\_RM\_RAW\_SUBCHANNEL ([see page 376](#))

### 1.1.10.3.6 SDiskCopyOptions::nVerifyBufferSectors Data Member

**C++**

```
int32 nVerifyBufferSectors;
```

**Description**

This value set the buffer size for the verify step. Valid value is 1-27. It will set the amount of sectors to read.

### 1.1.10.3.7 SDiskCopyOptions::nWriteMethod Data Member

**C++**

```
uint8 nWriteMethod;
```

**Description**

The write method for the target device. Allowed are BS\_WM\_DAO ([see page 506](#)) and BS\_WM.DAO96 ([see page 506](#)) where BS\_WM.DAO96 ([see page 506](#)) is only used if the ReadMode is BS\_RM\_RAW\_SUBCHANNEL ([see page 376](#)).

## 1.1.10.4 SDiskCopyOptions Methods

The methods of the SDiskCopyOptions class are listed here.

**Public Methods**

	<b>Name</b>	<b>Description</b>
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 27</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDiskCopyOptions.

### 1.1.10.4.1 SDiskCopyOptions::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL - sizeof(BS_BOOL) - sizeof(int32) -  
sizeof(BS_BOOL));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SDiskCopyOptions.

## 1.1.11 SDVDVideoOptions Structure

**C++**

```
struct SDVDVideoOptions {
    BS_BOOL Padding;
    BS_BOOL ForceUppercase;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains additional information for VideoDVD projects.

### 1.1.11.1 SDVDVideoOptions Members

The following tables list the members exposed by SDVDVideoOptions.

**Public Data Members**

	Name	Description
◆	ForceUppercase ( <a href="#">see page 29</a> )	Will make all files on the disc uppercase.
◆	Padding ( <a href="#">see page 29</a> )	IFO 32K padding. Some DVD authoring applications use this option to set a fixed place between IFO file and next file. If you have a VideoDVD image that was authored with this option you have to set this flag to TRUE to avoid playback problems.

**Public Methods**

	Name	Description
≡◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 29</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SDVDVideoOptions.
≡◆	SDVDVideoOptions ( <a href="#">see page 28</a> )	This is SDVDVideoOptions, a member of class SDVDVideoOptions.

### 1.1.11.2 SDVDVideoOptions::SDVDVideoOptions Constructor

**C++**

```
SDVDVideoOptions();
```

**Description**

This is SDVDVideoOptions, a member of class SDVDVideoOptions.

### 1.1.11.3 SDVDVideoOptions Data Members

The data members of the SDVDVideoOptions class are listed here.

**Public Data Members**

	Name	Description
◆	ForceUppercase (☞ see page 29)	Will make all files on the disc uppercase.
◆	Padding (☞ see page 29)	IFO 32K padding. Some DVD authoring applications use this option to set a fixed place between IFO file and next file. If you have a VideoDVD image that was authored with this option you have to set this flag to TRUE to avoid playback problems.

**1.1.11.3.1 SDVDVideoOptions::ForceUppercase Data Member****C++**

```
BS_BOOL ForceUppercase;
```

**Description**

Will make all files on the disc uppercase.

**1.1.11.3.2 SDVDVideoOptions::Padding Data Member****C++**

```
BS_BOOL Padding;
```

**Description**

IFO 32K padding. Some DVD authoring applications use this option to set a fixed place between IFO file and next file. If you have a VideoDVD image that was authored with this option you have to set this flag to TRUE to avoid playback problems.

**1.1.11.4 SDVDVideoOptions Methods**

The methods of the SDVDVideoOptions class are listed here.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING (☞ see page 29)	This is INSERT_STRUCTURE_PADDING, a member of class SDVDVideoOptions.

**1.1.11.4.1 SDVDVideoOptions::INSERT\_STRUCTURE\_PADDING Method****C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SDVDVideoOptions.

**1.1.12 SExtendedDeviceInformation Structure****C++**

```
struct SExtendedDeviceInformation {
    TCHAR lpszName[_MAX_PATH];
    TCHAR lpszRevision[_MAX_PATH];
    int32 nRegionCode;
    int32 nRegionCodeChangesLeft;
    TCHAR lpszLoaderType[_MAX_PATH];
    TCHAR lpszConnectionInterface[_MAX_PATH];
```

```

TCHAR lpszPhysicalInterface[_MAX_PATH];
int32 nNumberOfVolumeLevels;
int32 nBufferSize;
TCHAR lpszSerialNumber[_MAX_PATH];
int32 nReadRetryCount;
int32 nRegionCodeVendorResetsLeft;
time_t FirmwareCreationDate;
int32 nIdeTransferMode;
};

}

```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the extended information for the device.

### 1.1.12.1 SExtendedDeviceInformation Members

The following tables list the members exposed by SExtendedDeviceInformation.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	FirmwareCreationDate ( <a href="#">see page 31</a> )	The firmware date of the device.
◆	lpszConnectionInterface ( <a href="#">see page 32</a> )	The interface of the device. Like IDE, SATA, USB and so forth.
◆	lpszLoaderType ( <a href="#">see page 32</a> )	The type of the loader, like example “tray”.
◆	lpszName ( <a href="#">see page 32</a> )	The full name of the device.
◆	lpszPhysicalInterface ( <a href="#">see page 32</a> )	The physical drive interface like “ATAPI”.
◆	lpszRevision ( <a href="#">see page 32</a> )	The drive revision.
◆	lpszSerialNumber ( <a href="#">see page 32</a> )	The serial number of the device.
◆	nBufferSize ( <a href="#">see page 33</a> )	The device buffer size.
◆	nIdeTransferMode ( <a href="#">see page 33</a> )	The IDE transfer mode of the device. This is not available on all devices.
◆	nNumberOfVolumeLevels ( <a href="#">see page 33</a> )	The number of volume levels of the device.
◆	nReadRetryCount ( <a href="#">see page 33</a> )	The read retry count that is used by the drive if a read error occur.
◆	nRegionCode ( <a href="#">see page 33</a> )	The currently set region code of the device.
◆	nRegionCodeChangesLeft ( <a href="#">see page 33</a> )	The available region code changes of the device.
◆	nRegionCodeVendorResetsLeft ( <a href="#">see page 34</a> )	The available vendor region code changes of the device.

**Public Methods**

	<b>Name</b>	<b>Description</b>
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 34</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SExtendedDeviceInformation.
◆	SExtendedDeviceInformation ( <a href="#">see page 31</a> )	This is SExtendedDeviceInformation, a member of class SExtendedDeviceInformation.

## 1.1.12.2 SExtendedDeviceInformation::SExtendedDeviceInformation Constructor

C++

```
SExtendedDeviceInformation();
```

### Description

This is SExtendedDeviceInformation, a member of class SExtendedDeviceInformation.

## 1.1.12.3 SExtendedDeviceInformation Data Members

The data members of the SExtendedDeviceInformation class are listed here.

### Public Data Members

	Name	Description
◆	FirmwareCreationDate (↗ see page 31)	The firmware date of the device.
◆	IpszConnectionInterface (↗ see page 32)	The interface of the device. Like IDE, SATA, USB and so forth.
◆	IpszLoaderType (↗ see page 32)	The type of the loader, like example “tray”.
◆	IpszName (↗ see page 32)	The full name of the device.
◆	IpszPhysicalInterface (↗ see page 32)	The physical drive interface like “ATAPI”.
◆	IpszRevision (↗ see page 32)	The drive revision.
◆	IpszSerialNumber (↗ see page 32)	The serial number of the device.
◆	nBufferSize (↗ see page 33)	The device buffer size.
◆	nIdeTransferMode (↗ see page 33)	The IDE transfer mode of the device. This is not available on all devices.
◆	nNumberOfVolumeLevels (↗ see page 33)	The number of volume levels of the device.
◆	nReadRetryCount (↗ see page 33)	The read retry count that is used by the drive if a read error occur.
◆	nRegionCode (↗ see page 33)	The currently set region code of the device.
◆	nRegionCodeChangesLeft (↗ see page 33)	The available region code changes of the device.
◆	nRegionCodeVendorResetsLeft (↗ see page 34)	The available vendor region code changes of the device.

### 1.1.12.3.1 SExtendedDeviceInformation::FirmwareCreationDate Data Member

C++

```
time_t FirmwareCreationDate;
```

### Description

The firmware date of the device.

### 1.1.12.3.2 SExtendedDeviceInformation::lpszConnectionInterface Data Member

C++

```
TCHAR lpszConnectionInterface[_MAX_PATH];
```

**Description**

The interface of the device. Like IDE, SATA, USB and so forth.

### 1.1.12.3.3 SExtendedDeviceInformation::lpszLoaderType Data Member

C++

```
TCHAR lpszLoaderType[_MAX_PATH];
```

**Description**

The type of the loader, like example “tray”.

### 1.1.12.3.4 SExtendedDeviceInformation::lpszName Data Member

C++

```
TCHAR lpszName[_MAX_PATH];
```

**Description**

The full name of the device.

### 1.1.12.3.5 SExtendedDeviceInformation::lpszPhysicalInterface Data Member

C++

```
TCHAR lpszPhysicalInterface[_MAX_PATH];
```

**Description**

The physical drive interface like “ATAPI”.

### 1.1.12.3.6 SExtendedDeviceInformation::lpszRevision Data Member

C++

```
TCHAR lpszRevision[_MAX_PATH];
```

**Description**

The drive revision.

### 1.1.12.3.7 SExtendedDeviceInformation::lpszSerialNumber Data Member

C++

```
TCHAR lpszSerialNumber[_MAX_PATH];
```

**Description**

The serial number of the device.

### 1.1.12.3.8 SExtendedDeviceInformation::nBufferSize Data Member

C++

```
int32 nBufferSize;
```

**Description**

The device buffer size.

### 1.1.12.3.9 SExtendedDeviceInformation::nIdeTransferMode Data Member

C++

```
int32 nIdeTransferMode;
```

**Description**

The IDE transfer mode of the device. This is not available on all devices.

### 1.1.12.3.10 SExtendedDeviceInformation::nNumberOfVolumeLevels Data Member

C++

```
int32 nNumberOfVolumeLevels;
```

**Description**

The number of volume levels of the device.

### 1.1.12.3.11 SExtendedDeviceInformation::nReadRetryCount Data Member

C++

```
int32 nReadRetryCount;
```

**Description**

The read retry count that is used by the drive if a read error occur.

### 1.1.12.3.12 SExtendedDeviceInformation::nRegionCode Data Member

C++

```
int32 nRegionCode;
```

**Description**

The currently set region code of the device.

### 1.1.12.3.13 SExtendedDeviceInformation::nRegionCodeChangesLeft Data Member

C++

```
int32 nRegionCodeChangesLeft;
```

**Description**

The available region code changes of the device.

### 1.1.12.3.14 SExtendedDeviceInformation::nRegionCodeVendorResetsLeft Data Member

**C++**

```
int32 nRegionCodeVendorResetsLeft;
```

#### Description

The available vendor region code changes of the device.

## 1.1.12.4 SExtendedDeviceInformation Methods

The methods of the SExtendedDeviceInformation class are listed here.

#### Public Methods

	Name	Description
💡	INSERT_STRUCTURE_PADDING (↗ see page 34)	This is INSERT_STRUCTURE_PADDING, a member of class SExtendedDeviceInformation.

### 1.1.12.4.1 SExtendedDeviceInformation::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - sizeof(int32) - sizeof(time_t) - sizeof(int32));
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SExtendedDeviceInformation.

---

## 1.1.13 SFileAudioProperty Structure

**C++**

```
struct SFileAudioProperty {
    const TCHAR* lpszSourceFilePath;
    const TCHAR* lpszTitle;
    const TCHAR* Performer;
    const TCHAR* SongWriter;
    const TCHAR* Composer;
    const TCHAR* Arranger;
    const TCHAR* Message;
    const TCHAR* lpszMCN_ISRC;
    int nPause;
    const int32* pIndexes;
    int32 nIndexesLength;
    BS_BOOL bPauseInFrames;
};
```

#### File

IsoSDKDefinitions.h (↗ see page 517)

#### Description

IsoSDK sub class that contains information about CD-Text and track properties.

## 1.1.13.1 SFileAudioProperty Members

The following tables list the members exposed by SFileAudioProperty.

### Public Data Members

	Name	Description
◆	Arranger (see page 36)	The arranger of the track/disc (CD-TEXT).
◆	bPauseInFrames (see page 36)	The pause size before a track in frames. Ignored for disc.
◆	Composer (see page 36)	The composer of the track/disc (CD-TEXT).
◆	IpszMCN_ISRC (see page 36)	MCN or ISRC depending on IpszSourceFilePath (see page 36); TRACK = ISRC = International Standard Recording Code ( <a href="http://www.ifpi.org/">http://www.ifpi.org/</a> ); DISK = MCN = Media Catalog Number.
◆	IpszSourceFilePath (see page 36)	Pointer to the null-terminated string with the source path of the audio file which properties you want to change; or the root directory of audio files ("\" for audio projects and "\audio" for Mixed Mode projects) if you want to change the properties for a disk.
◆	IpszTitle (see page 37)	The title of the track/disc (CD-TEXT).
◆	Message (see page 37)	The message of the track/disc (CD-TEXT).
◆	nIndexesLength (see page 37)	Number of items in plndexes (see page 37).
◆	nPause (see page 37)	The pause size before a track in seconds. Ignored for disc.
◆	Performer (see page 37)	The performer of the track/disc (CD-TEXT).
◆	plndexes (see page 37)	A pointer to a n array of indexes inside track. This are the subindexes for the Track only, not for the disc.
◆	SongWriter (see page 38)	The song writer of the track/disc (CD-TEXT).

### Public Methods

	Name	Description
◆	INSERT_STRUCTURE_PADDING (see page 38)	This is INSERT_STRUCTURE_PADDING, a member of class SFileAudioProperty.
◆	SFileAudioProperty (see page 35)	This is SFileAudioProperty, a member of class SFileAudioProperty.

## 1.1.13.2 SFileAudioProperty::SFileAudioProperty Constructor

### C++

```
SFileAudioProperty();
```

### Description

This is SFileAudioProperty, a member of class SFileAudioProperty.

## 1.1.13.3 SFileAudioProperty Data Members

The data members of the SFileAudioProperty class are listed here.

### Public Data Members

	Name	Description
◆	Arranger (see page 36)	The arranger of the track/disc (CD-TEXT).
◆	bPauseInFrames (see page 36)	The pause size before a track in frames. Ignored for disc.
◆	Composer (see page 36)	The composer of the track/disc (CD-TEXT).
◆	IpszMCN_ISRC (see page 36)	MCN or ISRC depending on IpszSourceFilePath (see page 36); TRACK = ISRC = International Standard Recording Code ( <a href="http://www.ifpi.org/">http://www.ifpi.org/</a> ); DISK = MCN = Media Catalog Number.

◆	IpszSourceFilePath (see page 36)	Pointer to the null-terminated string with the source path of the audio file which properties you want to change; or the root directory of audio files ("\" for audio projects and "\audio" for Mixed Mode projects) if you want to change the properties for a disk.
◆	IpszTitle (see page 37)	The title of the track/disc (CD-TEXT).
◆	Message (see page 37)	The message of the track/disc (CD-TEXT).
◆	nIndexesLength (see page 37)	Number of items in plndexes (see page 37).
◆	nPause (see page 37)	The pause size before a track in seconds. Ignored for disc.
◆	Performer (see page 37)	The performer of the track/disc (CD-TEXT).
◆	plndexes (see page 37)	A pointer to a n array of indexes inside track. This are the subindexes for the Track only, not for the disc.
◆	SongWriter (see page 38)	The song writer of the track/disc (CD-TEXT).

### 1.1.13.3.1 SFileAudioProperty::Arranger Data Member

C++

```
const TCHAR* Arranger;
```

**Description**

The arranger of the track/disc (CD-TEXT).

### 1.1.13.3.2 SFileAudioProperty::bPauseInFrames Data Member

C++

```
BS_BOOL bPauseInFrames;
```

**Description**

The pause size before a track in frames. Ignored for disc.

### 1.1.13.3.3 SFileAudioProperty::Composer Data Member

C++

```
const TCHAR* Composer;
```

**Description**

The composer of the track/disc (CD-TEXT).

### 1.1.13.3.4 SFileAudioProperty::lpszMCN\_ISRC Data Member

C++

```
const TCHAR* lpszMCN_ISRC;
```

**Description**

MCN or ISRC depending on IpszSourceFilePath (see page 36); TRACK = ISRC = International Standard Recording Code (<http://www.ifpi.org/>); DISK = MCN = Media Catalog Number.

### 1.1.13.3.5 SFileAudioProperty::lpszSourceFilePath Data Member

C++

```
const TCHAR* lpszSourceFilePath;
```

**Description**

Pointer to the null-terminated string with the source path of the audio file which properties you want to change; or the root

directory of audio files ("\" for audio projects and "\audio" for Mixed Mode projects) if you want to change the properties for a disk.

### 1.1.13.3.6 SFileAudioProperty::lpszTitle Data Member

C++

```
const TCHAR* lpszTitle;
```

Description

The title of the track/disc (CD-TEXT).

### 1.1.13.3.7 SFileAudioProperty::Message Data Member

C++

```
const TCHAR* Message;
```

Description

The message of the track/disc (CD-TEXT).

### 1.1.13.3.8 SFileAudioProperty::nIndexesLength Data Member

C++

```
int32 nIndexesLength;
```

Description

Number of items in pIndexes (see page 37).

### 1.1.13.3.9 SFileAudioProperty::nPause Data Member

C++

```
int nPause;
```

Description

The pause size before a track in seconds. Ignored for disc.

### 1.1.13.3.10 SFileAudioProperty::Performer Data Member

C++

```
const TCHAR* Performer;
```

Description

The performer of the track/disc (CD-TEXT).

### 1.1.13.3.11 SFileAudioProperty::pIndexes Data Member

C++

```
const int32* pIndexes;
```

Description

A pointer to a n array of indexes inside track. This are the subindexes for the Track only, not for the disc.

### 1.1.13.3.12 SFileDialogProperty::SongWriter Data Member

**C++**

```
const TCHAR* SongWriter;
```

**Description**

The song writer of the track/disc (CD-TEXT).

### 1.1.13.4 SFileDialogProperty Methods

The methods of the SFileDialogProperty class are listed here.

**Public Methods**

	Name	Description
💡	INSERT_STRUCTURE_PADDING (see page 38)	This is INSERT_STRUCTURE_PADDING, a member of class SFileDialogProperty.

#### 1.1.13.4.1 SFileDialogProperty::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_BIG - sizeof(BS_BOOL));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileDialogProperty.

## 1.1.14 SFileDialogTime Structure

**C++**

```
struct SFileDialogTime {
    uint8 nYear;
    uint8 nMonth;
    uint8 nDay;
    uint8 nHour;
    uint8 nMinute;
    uint8 nSecond;
};
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK sub class that contains the information about the time & date of the specific file.

### 1.1.14.1 SFileDialogTime Members

The following tables list the members exposed by SFileDialogTime.

**Public Data Members**

	Name	Description
💡	nDay (see page 39)	Day of month (1....31)
💡	nHour (see page 39)	Hours (0...23)

◆	nMinute ( <a href="#">see page 39</a> )	Minutes (0...59)
◆	nMonth ( <a href="#">see page 39</a> )	Month (1..12)
◆	nSecond ( <a href="#">see page 40</a> )	Seconds (0...59)
◆	nYear ( <a href="#">see page 40</a> )	Years since 1900

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 40</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileDateTime.

## 1.1.14.2 SFileDateTime Data Members

The data members of the SFileDateTime class are listed here.

**Public Data Members**

	Name	Description
◆	nDay ( <a href="#">see page 39</a> )	Day of month (1....31)
◆	nHour ( <a href="#">see page 39</a> )	Hours (0...23)
◆	nMinute ( <a href="#">see page 39</a> )	Minutes (0...59)
◆	nMonth ( <a href="#">see page 39</a> )	Month (1..12)
◆	nSecond ( <a href="#">see page 40</a> )	Seconds (0...59)
◆	nYear ( <a href="#">see page 40</a> )	Years since 1900

### 1.1.14.2.1 SFileDateTime::nDay Data Member

**C++**

```
uint8 nDay;
```

**Description**

Day of month (1....31)

### 1.1.14.2.2 SFileDateTime::nHour Data Member

**C++**

```
uint8 nHour;
```

**Description**

Hours (0...23)

### 1.1.14.2.3 SFileDateTime::nMinute Data Member

**C++**

```
uint8 nMinute;
```

**Description**

Minutes (0...59)

### 1.1.14.2.4 SFileDateTime::nMonth Data Member

**C++**

```
uint8 nMonth;
```

**Description**

Month (1..12)

**1.1.14.2.5 SFileDialogTime::nSecond Data Member****C++**

```
uint8 nSecond;
```

**Description**

Seconds (0...59)

**1.1.14.2.6 SFileDialogTime::nYear Data Member****C++**

```
uint8 nYear;
```

**Description**

Years since 1900

**1.1.14.3 SFileDialogTime Methods**

The methods of the SFileDialogTime class are listed here.

**Public Methods**

	Name	Description
💡	INSERT_STRUCTURE_PADDING (↗ see page 40)	This is INSERT_STRUCTURE_PADDING, a member of class SFileDialogTime.

**1.1.14.3.1 SFileDialogTime::INSERT\_STRUCTURE\_PADDING Method****C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileDialogTime.

---

**1.1.15 SFileEntry Structure****C++**

```
struct SFileEntry {
    TCHAR lpszFileName[_MAX_PATH];
    TCHAR lpszFilePath[_MAX_PATH];
    TCHAR lpszFileOrigin[_MAX_PATH];
    int32 nAddress;
    int64 nFileSize;
    SFileDialogTime cDateTime;
    int32 nAttrib;
    SFileDialogTime cCreationTime;
    SFileDialogTime cAccessTime;
    void* pUserParam;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the file to be used.

### 1.1.15.1 SFileEntry Members

The following tables list the members exposed by SFileEntry.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
❖	cAccessTime ( <a href="#">see page 42</a> )	The last access date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	cCreationTime ( <a href="#">see page 42</a> )	The creation date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	cDateTime ( <a href="#">see page 42</a> )	The last modified date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	lpszFileName ( <a href="#">see page 42</a> )	The name of the directory entry (null terminated).
❖	lpszFileOrigin ( <a href="#">see page 42</a> )	The path of the file on HDD. If the file is imported from the disc session, this field is NULL.
❖	lpszFilePath ( <a href="#">see page 43</a> )	The path to the containing directory of this entry (null terminated).
❖	nAddress ( <a href="#">see page 43</a> )	This is the file descriptor address of the file (LBA).
❖	nAttrib ( <a href="#">see page 43</a> )	Defines the file attribute of a file to add. Possible values are: BS_FA_READONLY ( <a href="#">see page 335</a> ), BS_FA_HIDDEN ( <a href="#">see page 334</a> ), BS_FA_ADVANCED_HIDDEN ( <a href="#">see page 333</a> ), BS_FA_SYSTEM ( <a href="#">see page 335</a> ), BS_FA_DIRECTORY ( <a href="#">see page 334</a> ), BS_FA_ARCHIVE ( <a href="#">see page 334</a> )
❖	nFileSize ( <a href="#">see page 43</a> )	The size of the file in bytes.
❖	pUserParam ( <a href="#">see page 43</a> )	This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.

**Public Methods**

	<b>Name</b>	<b>Description</b>
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 44</a> )	This is the overview for the INSERT_STRUCTURE_PADDING method overload.

### 1.1.15.2 SFileEntry Data Members

The data members of the SFileEntry class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
❖	cAccessTime ( <a href="#">see page 42</a> )	The last access date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	cCreationTime ( <a href="#">see page 42</a> )	The creation date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	cDateTime ( <a href="#">see page 42</a> )	The last modified date of the file according the file system. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
❖	lpszFileName ( <a href="#">see page 42</a> )	The name of the directory entry (null terminated).

❖	IpszFileOrigin (see page 42)	The path of the file on HDD. If the file is imported from the disc session, this field is NULL.
❖	IpszFilePath (see page 43)	The path to the containing directory of this entry (null terminated).
❖	nAddress (see page 43)	This is the file descriptor address of the file (LBA).
❖	nAttrib (see page 43)	Defines the file attribute of a file to add. Possible values are: BS_FA_READONLY (see page 335), BS_FA_HIDDEN (see page 334), BS_FA_ADVANCED_HIDDEN (see page 333), BS_FA_SYSTEM (see page 335), BS_FA_DIRECTORY (see page 334), BS_FA_ARCHIVE (see page 334)
❖	nFileSize (see page 43)	The size of the file in bytes.
❖	pUserParam (see page 43)	This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.

### 1.1.15.2.1 SFileEntry::cAccessTime Data Member

C++

```
SFileDateTime cAccessTime;
```

#### Description

The last access date of the file according the file system. Use a SFileDateTime (see page 38) structure as value.

### 1.1.15.2.2 SFileEntry::cCreationTime Data Member

C++

```
SFileDateTime cCreationTime;
```

#### Description

The creation date of the file according the file system. Use a SFileDateTime (see page 38) structure as value.

### 1.1.15.2.3 SFileEntry::cDateTime Data Member

C++

```
SFileDateTime cDateTime;
```

#### Description

The last modified date of the file according the file system. Use a SFileDateTime (see page 38) structure as value.

### 1.1.15.2.4 SFileEntry::lpszFileName Data Member

C++

```
TCHAR lpszFileName[_MAX_PATH];
```

#### Description

The name of the directory entry (null terminated).

### 1.1.15.2.5 SFileEntry::lpszFileOrigin Data Member

C++

```
TCHAR lpszFileOrigin[_MAX_PATH];
```

**Description**

The path of the file on HDD. If the file is imported from the disc session, this field is NULL.

### 1.1.15.2.6 SFileEntry::lpszFilePath Data Member

**C++**

```
TCHAR lpszFilePath[_MAX_PATH];
```

**Description**

The path to the containing directory of this entry (null terminated).

### 1.1.15.2.7 SFileEntry::nAddress Data Member

**C++**

```
int32 nAddress;
```

**Description**

This is the file descriptor address of the file (LBA).

### 1.1.15.2.8 SFileEntry::nAttrib Data Member

**C++**

```
int32 nAttrib;
```

**Description**

Defines the file attribute of a file to add. Possible values are:

BS\_FA\_READONLY (see page 335),

BS\_FA\_HIDDEN (see page 334),

BS\_FA\_ADVANCED\_HIDDEN (see page 333),

BS\_FA\_SYSTEM (see page 335),

BS\_FA\_DIRECTORY (see page 334),

BS\_FA\_ARCHIVE (see page 334)

### 1.1.15.2.9 SFileEntry::nFileSize Data Member

**C++**

```
int64 nFileSize;
```

**Description**

The size of the file in bytes.

### 1.1.15.2.10 SFileEntry::pUserParam Data Member

**C++**

```
void* pUserParam;
```

**Description**

This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.

## 1.1.15.3 SFileEntry Methods

The methods of the SFileEntry class are listed here.

### Public Methods

	Name	Description
≡♦	INSERT_STRUCTURE_PADDING ( <a href="#">see page 44</a> )	This is the overview for the INSERT_STRUCTURE_PADDING method overload.

### 1.1.15.3.1 INSERT\_STRUCTURE\_PADDING Method

This is the overview for the INSERT\_STRUCTURE\_PADDING method overload.

#### Overload List

	Name	Description
≡♦	SFileEntry::INSERT_STRUCTURE_PADDING (STRUCTURE_PADDING_STRING_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void*)) ( <a href="#">see page 44</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileEntry.
≡♦	SFileEntry::INSERT_STRUCTURE_PADDING (STRUCTURE_PADDING_STRING_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void*) - 8) ( <a href="#">see page 44</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileEntry.

#### 1.1.15.3.1.1 SFileEntry::INSERT\_STRUCTURE\_PADDING Method

**(STRUCTURE\_PADDING\_STRING\_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void\*))**

##### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void*));
```

##### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileEntry.

#### 1.1.15.3.1.2 SFileEntry::INSERT\_STRUCTURE\_PADDING Method

**(STRUCTURE\_PADDING\_STRING\_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void\*) - 8)**

##### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - sizeof(SFileDialogTime) - sizeof(SFileDialogTime) - sizeof(void*) - 8);
```

##### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileEntry.

---

## 1.1.16 SFileTimeEx Structure [new](#)

##### C++

```
struct SFileTimeEx {
    BS_BOOL UseCreationDateTime;
```

```

SFileTypeDateTime CreationDateTime;
BS_BOOL UseModificationDateTime;
SFileTypeDateTime ModificationDateTime;
BS_BOOL UseLastAccessDateTime;
SFileTypeDateTime LastAccessDateTime;
BS_BOOL UseCustomTimes;
};

```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.

You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes ([see page 211](#)) method.

This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.

## 1.1.16.1 SFileTypeEx Members

The following tables list the members exposed by SFileTypeEx.

**Public Data Members**

	Name	Description
 <a href="#">CreationDateTime</a> ( <a href="#">see page 46</a> )	The CreationDateTime value.	
 <a href="#">LastAccessDateTime</a> ( <a href="#">see page 46</a> )	The LastAccessDateTime value.	
 <a href="#">ModificationDateTime</a> ( <a href="#">see page 46</a> )	The ModificationDateTime value.	
 <a href="#">UseCreationDateTime</a> ( <a href="#">see page 46</a> )	Switch to enable the CreationDateTime ( <a href="#">see page 46</a> ) value. Only enabled if the UseCustomTimes ( <a href="#">see page 47</a> ) switch is also set to true.	
 <a href="#">UseCustomTimes</a> ( <a href="#">see page 47</a> )	The global switch to enable the custom date / time information for files, directories and file systems. If the FS only allow one date, the Modification date / time info is used.	
 <a href="#">UseLastAccessDateTime</a> ( <a href="#">see page 47</a> )	Switch to enable the LastAccessDateTime ( <a href="#">see page 46</a> ) value. Only enabled if the UseCustomTimes ( <a href="#">see page 47</a> ) switch is also set to true.	
 <a href="#">UseModificationDateTime</a> ( <a href="#">see page 47</a> )	Switch to enable the ModificationDateTime ( <a href="#">see page 46</a> ) value. Only enabled if the UseCustomTimes ( <a href="#">see page 47</a> ) switch is also set to true.	

**Public Methods**

	Name	Description
 <a href="#">SFileTypeEx</a> ( <a href="#">see page 45</a> )		

## 1.1.16.2 SFileTypeEx::SFileTypeEx Constructor

**C++**

```
SFileTypeEx();
```

## 1.1.16.3 SFileTimeEx Data Members

The data members of the SFileTimeEx class are listed here.

### Public Data Members

	Name	Description
↳ new	CreationDateTime (see page 46)	The CreationDateTime value.
↳ new	LastAccessDateTime (see page 46)	The LastAccessDateTime value.
↳ new	ModificationDateTime (see page 46)	The ModificationDateTime value.
↳ new	UseCreationDateTime (see page 46)	Switch to enable the CreationDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.
↳ new	UseCustomTimes (see page 47)	The global switch to enable the custom date / time information for files, directories and file systems. If the FS only allow one date, the Modification date / time info is used.
↳ new	UseLastAccessDateTime (see page 47)	Switch to enable the LastAccessDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.
↳ new	UseModificationDateTime (see page 47)	Switch to enable the ModificationDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.

### 1.1.16.3.1 SFileTimeEx::CreationDateTime Data Member new

#### C++

```
SFileDateTime CreationDateTime;
```

#### Description

The CreationDateTime value.

### 1.1.16.3.2 SFileTimeEx::LastAccessDateTime Data Member new

#### C++

```
SFileDateTime LastAccessDateTime;
```

#### Description

The LastAccessDateTime value.

### 1.1.16.3.3 SFileTimeEx::ModificationDateTime Data Member new

#### C++

```
SFileDateTime ModificationDateTime;
```

#### Description

The ModificationDateTime value.

### 1.1.16.3.4 SFileTimeEx::UseCreationDateTime Data Member new

#### C++

```
BS_BOOL UseCreationDateTime;
```

**Description**

Switch to enable the CreationDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.

**1.1.16.3.5 SFileTypeEx::UseCustomTimes Data Member new****C++**

```
BS_BOOL UseCustomTimes;
```

**Description**

The global switch to enable the custom date / time information for files, directories and file systems. If the FS only allow one date, the Modification date / time info is used.

**1.1.16.3.6 SFileTypeEx::UseLastAccessDateTime Data Member new****C++**

```
BS_BOOL UseLastAccessDateTime;
```

**Description**

Switch to enable the LastAccessDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.

**1.1.16.3.7 SFileTypeEx::UseModificationDateTime Data Member new****C++**

```
BS_BOOL UseModificationDateTime;
```

**Description**

Switch to enable the ModificationDateTime (see page 46) value. Only enabled if the UseCustomTimes (see page 47) switch is also set to true.

---

**1.1.17 SFileToAdd Structure****C++**

```
struct SFileToAdd {
    const TCHAR* lpszSourceFilePath;
    const TCHAR* lpszDestinationPath;
    const TCHAR* lpszFileName;
    BS_BOOL bVideoFile;
    int32 nSavePath;
};
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK sub class that contains the information about the file you want to add to the project.

**Remarks**

The value lpszDestinationPath (see page 49) is basing on the rootDirectory. The rootDirectory is always "\\\" expected MixedMode project. Here the root for audio is "\\audio" and for data files "\\data".

## 1.1.17.1 SFileToAdd Members

The following tables list the members exposed by SFileToAdd.

### Public Data Members

	Name	Description
◆	bVideoFile (see page 49)	This value says whether the file is a video file. BS_TRUE (see page 499) describes a video file, BS_FALSE (see page 335) marks all other types.
◆	IpszDestinationPath (see page 49)	Pointer to the null-terminated string with the path of the destination folder of the project.
◆	IpszFileName (see page 49)	Pointer to the null-terminated string with the name of the file that you want to add to the project. <b>Deprecated!</b> Please leave this value NULL
◆	IpszSourceFilePath (see page 49)	Pointer to the null-terminated string with the path and file name of the file you want to add to the project.
◆	nSavePath (see page 49)	This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed: BS_DONT_SAVE_PATH (see page 326), BS_WHOLE_PATH (see page 506), BS_PARENTDIR_ONLY (see page 367)

### Public Methods

	Name	Description
≡	INSERT_STRUCTURE_PADDING (see page 50)	This is INSERT_STRUCTURE_PADDING, a member of class SFileToAdd.
≡	SFileToAdd (see page 48)	This is SFileToAdd, a member of class SFileToAdd.

## 1.1.17.2 SFileToAdd::SFileToAdd Constructor

### C++

```
SFileToAdd();
```

### Description

This is SFileToAdd, a member of class SFileToAdd.

## 1.1.17.3 SFileToAdd Data Members

The data members of the SFileToAdd class are listed here.

### Public Data Members

	Name	Description
◆	bVideoFile (see page 49)	This value says whether the file is a video file. BS_TRUE (see page 499) describes a video file, BS_FALSE (see page 335) marks all other types.
◆	IpszDestinationPath (see page 49)	Pointer to the null-terminated string with the path of the destination folder of the project.
◆	IpszFileName (see page 49)	Pointer to the null-terminated string with the name of the file that you want to add to the project. <b>Deprecated!</b> Please leave this value NULL

❖	lpszSourceFilePath ( <a href="#">see page 49</a> )	Pointer to the null-terminated string with the path and file name of the file you want to add to the project.
❖	nSavePath ( <a href="#">see page 49</a> )	This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed: BS_DONT_SAVE_PATH ( <a href="#">see page 326</a> ), BS_WHOLE_PATH ( <a href="#">see page 506</a> ), BS_PARENTDIR_ONLY ( <a href="#">see page 367</a> )

### 1.1.17.3.1 SFileToAdd::bVideoFile Data Member

C++

```
BS_BOOL bVideoFile;
```

**Description**

This value says whether the file is a video file. BS\_TRUE ([see page 499](#)) describes a video file, BS\_FALSE ([see page 335](#)) marks all other types.

### 1.1.17.3.2 SFileToAdd::lpszDestinationPath Data Member

C++

```
const TCHAR* lpszDestinationPath;
```

**Description**

Pointer to the null-terminated string with the path of the destination folder of the project.

### 1.1.17.3.3 SFileToAdd::lpszFileName Data Member

C++

```
const TCHAR* lpszFileName;
```

**Description**

Pointer to the null-terminated string with the name of the file that you want to add to the project.

**Deprecated!** Please leave this value NULL

### 1.1.17.3.4 SFileToAdd::lpszSourceFilePath Data Member

C++

```
const TCHAR* lpszSourceFilePath;
```

**Description**

Pointer to the null-terminated string with the path and file name of the file you want to add to the project.

### 1.1.17.3.5 SFileToAdd::nSavePath Data Member

C++

```
int32 nSavePath;
```

**Description**

This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed:

BS\_DONT\_SAVE\_PATH (↗ see page 326),  
 BS\_WHOLE\_PATH (↗ see page 506),  
 BS\_PARENTDIR\_ONLY (↗ see page 367)

## 1.1.17.4 SFileToAdd Methods

The methods of the SFileToAdd class are listed here.

### Public Methods

	Name	Description
💡	INSERT_STRUCTURE_PADDING (↗ see page 50)	This is INSERT_STRUCTURE_PADDING, a member of class SFileToAdd.

### 1.1.17.4.1 SFileToAdd::INSERT\_STRUCTURE\_PADDING Method

#### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileToAdd.

## 1.1.18 SFileToAddEx Structure

#### C++

```
struct SFileToAddEx {
    const TCHAR* lpszSourceFilePath;
    const TCHAR* lpszDestinationPath;
    const TCHAR* lpszFileName;
    int32 nSavePath;
    int64 nFileOffset;
    int64 nSegmentSize;
};
```

#### File

IsoSDKDefinitions.h (↗ see page 517)

#### Description

IsoSDK sub class that contains the information about the file segment you want to add to the project.

#### Remarks

The value lpszDestinationPath (↗ see page 51) is basing on the rootDirectory. The rootDirectory is always "\\\" expected MixedMode project. Here the root for audio is "\\audio" and for data files "\\data".

## 1.1.18.1 SFileToAddEx Members

The following tables list the members exposed by SFileToAddEx.

### Public Data Members

	Name	Description
💡	lpszDestinationPath (↗ see page 51)	Pointer to the null-terminated string with the path of the destination folder of the project.

❖	IpszFileName ( <a href="#">see page 52</a> )	Pointer to the null-terminated string with the name of the file that you want to add to the project.
❖	IpszSourceFilePath ( <a href="#">see page 52</a> )	Pointer to the null-terminated string with the path of the folder containing the file you want to add to the project.
❖	nFileOffset ( <a href="#">see page 52</a> )	The start position of the segment in the file in bytes.
❖	nSavePath ( <a href="#">see page 52</a> )	This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed: BS_DONT_SAVE_PATH ( <a href="#">see page 326</a> ), BS_WHOLE_PATH ( <a href="#">see page 506</a> ), BS_PARENTDIR_ONLY ( <a href="#">see page 367</a> )
❖	nSegmentSize ( <a href="#">see page 52</a> )	The size of the segment in bytes.

**Public Methods**

	Name	Description
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 53</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileToAddEx.
❖	SFileToAddEx ( <a href="#">see page 51</a> )	This is SFileToAddEx, a member of class SFileToAddEx.

## 1.1.18.2 SFileToAddEx::SFileToAddEx Constructor

**C++**`SFileToAddEx( ) ;`**Description**

This is SFileToAddEx, a member of class SFileToAddEx.

## 1.1.18.3 SFileToAddEx Data Members

The data members of the SFileToAddEx class are listed here.

**Public Data Members**

	Name	Description
❖	IpszDestinationPath ( <a href="#">see page 51</a> )	Pointer to the null-terminated string with the path of the destination folder of the project.
❖	IpszFileName ( <a href="#">see page 52</a> )	Pointer to the null-terminated string with the name of the file that you want to add to the project.
❖	IpszSourceFilePath ( <a href="#">see page 52</a> )	Pointer to the null-terminated string with the path of the folder containing the file you want to add to the project.
❖	nFileOffset ( <a href="#">see page 52</a> )	The start position of the segment in the file in bytes.
❖	nSavePath ( <a href="#">see page 52</a> )	This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed: BS_DONT_SAVE_PATH ( <a href="#">see page 326</a> ), BS_WHOLE_PATH ( <a href="#">see page 506</a> ), BS_PARENTDIR_ONLY ( <a href="#">see page 367</a> )
❖	nSegmentSize ( <a href="#">see page 52</a> )	The size of the segment in bytes.

### 1.1.18.3.1 SFileToAddEx::IpszDestinationPath Data Member

**C++**`const TCHAR* lpszDestinationPath;`**Description**

Pointer to the null-terminated string with the path of the destination folder of the project.

### 1.1.18.3.2 SFileToAddEx::lpszFileName Data Member

C++

```
const TCHAR* lpszFileName;
```

Description

Pointer to the null-terminated string with the name of the file that you want to add to the project.

### 1.1.18.3.3 SFileToAddEx::lpszSourceFilePath Data Member

C++

```
const TCHAR* lpszSourceFilePath;
```

Description

Pointer to the null-terminated string with the path of the folder containing the file you want to add to the project.

### 1.1.18.3.4 SFileToAddEx::nFileOffset Data Member

C++

```
int64 nFileOffset;
```

Description

The start position of the segment in the file in bytes.

### 1.1.18.3.5 SFileToAddEx::nSavePath Data Member

C++

```
int32 nSavePath;
```

Description

This argument states if and how the super ordinate directory is added as a folder to the project. The following values are allowed:

BS\_DONT\_SAVE\_PATH (see page 326),

BS\_WHOLE\_PATH (see page 506),

BS\_PARENTDIR\_ONLY (see page 367)

### 1.1.18.3.6 SFileToAddEx::nSegmentSize Data Member

C++

```
int64 nSegmentSize;
```

Description

The size of the segment in bytes.

## 1.1.18.4 SFileToAddEx Methods

The methods of the SFileToAddEx class are listed here.

**Public Methods**

	Name	Description
☞	INSERT_STRUCTURE_PADDING (☞ see page 53)	This is INSERT_STRUCTURE_PADDING, a member of class SFileToAddEx.

**1.1.18.4.1 SFileToAddEx::INSERT\_STRUCTURE\_PADDING Method****C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileToAddEx.

**1.1.19 SFileToRemove Structure****C++**

```
struct SFileToRemove {
    const TCHAR* lpszFile;
    const TCHAR* lpszDestinationPath;
};
```

**File**

IsoSDKDefinitions.h (☞ see page 517)

**Description**

IsoSDK sub class that contains the information about the file to be removed.

**1.1.19.1 SFileToRemove Members**

The following tables list the members exposed by SFileToRemove.

**Public Data Members**

	Name	Description
☞	lpszDestinationPath (☞ see page 54)	Path to the file in the project (null terminated) / without file name.
☞	lpszFile (☞ see page 54)	File name (null terminated) / without path information.

**Public Methods**

	Name	Description
☞	INSERT_STRUCTURE_PADDING (☞ see page 54)	This is INSERT_STRUCTURE_PADDING, a member of class SFileToRemove.
☞	SFileToRemove (☞ see page 53)	This is SFileToRemove, a member of class SFileToRemove.

**1.1.19.2 SFileToRemove::SFileToRemove Constructor****C++**

```
SFileToRemove();
```

**Description**

This is SFileToRemove, a member of class SFileToRemove.

### 1.1.19.3 SFileToRemove Data Members

The data members of the SFileToRemove class are listed here.

#### Public Data Members

	Name	Description
◆	lpszDestinationPath ( <a href="#">see page 54</a> )	Path to the file in the project (null terminated) / without file name.
◆	lpszFile ( <a href="#">see page 54</a> )	File name (null terminated) / without path information.

#### 1.1.19.3.1 SFileToRemove::lpszDestinationPath Data Member

##### C++

```
const TCHAR* lpszDestinationPath;
```

##### Description

Path to the file in the project (null terminated) / without file name.

#### 1.1.19.3.2 SFileToRemove::lpszFile Data Member

##### C++

```
const TCHAR* lpszFile;
```

##### Description

File name (null terminated) / without path information.

### 1.1.19.4 SFileToRemove Methods

The methods of the SFileToRemove class are listed here.

#### Public Methods

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 54</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileToRemove.

#### 1.1.19.4.1 SFileToRemove::INSERT\_STRUCTURE\_PADDING Method

##### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

##### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileToRemove.

---

### 1.1.20 SFileToRename Structure

##### C++

```
struct SFileToRename {
    const TCHAR* lpszSourcePath;
    const TCHAR* lpszDestinationPath;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class contains the information about the file to be renamed.

## 1.1.20.1 SFileToRename Members

The following tables list the members exposed by SFileToRename.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	IpszDestinationPath ( <a href="#">see page 55</a> )	Pointer to the null-terminated string with the destination path of the file you want to rename/move.
◆	IpszSourcePath ( <a href="#">see page 56</a> )	Pointer to the null-terminated string with the source path of the file you want to rename/move.

**Public Methods**

	<b>Name</b>	<b>Description</b>
+=	INSERT_STRUCTURE_PADDING ( <a href="#">see page 56</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SFileToRename.
+=	SFileToRename ( <a href="#">see page 55</a> )	This is SFileToRename, a member of class SFileToRename.

## 1.1.20.2 SFileToRename::SFileToRename Constructor

**C++**

```
SFileToRename( );
```

**Description**

This is SFileToRename, a member of class SFileToRename.

## 1.1.20.3 SFileToRename Data Members

The data members of the SFileToRename class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	IpszDestinationPath ( <a href="#">see page 55</a> )	Pointer to the null-terminated string with the destination path of the file you want to rename/move.
◆	IpszSourcePath ( <a href="#">see page 56</a> )	Pointer to the null-terminated string with the source path of the file you want to rename/move.

### 1.1.20.3.1 SFileToRename::IpszDestinationPath Data Member

**C++**

```
const TCHAR* lpszDestinationPath;
```

**Description**

Pointer to the null-terminated string with the destination path of the file you want to rename/move.

### 1.1.20.3.2 SFileToRename::lpszSourcePath Data Member

**C++**

```
const TCHAR* lpszSourcePath;
```

#### Description

Pointer to the null-terminated string with the source path of the file you want to rename/move.

### 1.1.20.4 SFileToRename Methods

The methods of the SFileToRename class are listed here.

#### Public Methods

	Name	Description
!	INSERT_STRUCTURE_PADDING (see page 56)	This is INSERT_STRUCTURE_PADDING, a member of class SFileToRename.

### 1.1.20.4.1 SFileToRename::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SFileToRename.

## 1.1.21 SISOInfoEx Structure

**C++**

```
struct SISOInfoEx {
    TCHAR ISOAbstractIdentifier[36];
    TCHAR ISOApplicationIdentifier[128];
    TCHAR ISOBiblioIdentifier[36];
    TCHAR ISOCopyrightIdentifier[36];
    TCHAR ISODataPreparerIdentifier[128];
    TCHAR ISOPublisherIdentifier[128];
    TCHAR ISOSetIdentifier[128];
    TCHAR ISOSystemIdentifier[31];
    BS_BOOL ISOAddSuffix;
    int32 ISOLevel;
    BS_BOOL ISOAllowManyDirectories;
    BS_BOOL ISOAllowLowercaseNames;
    BS_BOOL ISOAllowLongISO9660Names;
    BS_BOOL ISOAllowLongJolietNames;
    BS_BOOL ISOUseCreationDateTime;
    SfileDateTime ISOCreationDateTime;
    BS_BOOL ISOUseModificationDateTime;
    SfileDateTime ISOModificationDateTime;
    BS_BOOL ISOUseExpirationDateTime;
    SfileDateTime ISOExpirationDateTime;
    BS_BOOL ISOUseEffectiveDateTime;
    SfileDateTime ISOEffectiveDateTime;
};
```

#### File

IsoSDKDefinitions.h (see page 517)

## Description

This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.

You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes ([see page 211](#)) method.

This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.

## 1.1.21.1 SISOInfoEx Members

The following tables list the members exposed by SISOInfoEx.

### Public Data Members

	Name	Description
◆	ISOAbstractFileIdentifier ( <a href="#">see page 60</a> )	The Volume Abstract File Identifier field of the ISO/Joliet image (35 characters maximum). For detailed information see: ISO 9660.
◆	ISOAddSuffix ( <a href="#">see page 60</a> )	Get/Set the bool value to enable ISO version number extension (1). The ISO 9660 specification requires that each filename include a ;x version number as the suffix. A full filename in ISO 9660 has three parts: <name>.<extension>;<version> Presumably the version part was designed to allow multiple versions of the same file to coexist.
◆	ISOAllowLongISO9660Names ( <a href="#">see page 60</a> )	If enabled (BS_TRUE ( <a href="#">see page 499</a> )), allows long file and directory names up to 207 ASCII characters in ISO9660 file system. If options is disabled (BS_FALSE ( <a href="#">see page 335</a> )), the ISO Level restrictions on file name length will be used.
◆	ISOAllowLongJolietNames ( <a href="#">see page 60</a> )	If enabled (BS_TRUE ( <a href="#">see page 499</a> )), allows long file and directory names up to 103 UCS-2 characters in Joliet extension to ISO9660 file system. If options is disabled (BS_FALSE ( <a href="#">see page 335</a> )), the max file name length will be limited by 64 UCS-2 characters.
◆	ISOAllowLowercaseNames ( <a href="#">see page 60</a> )	BS_TRUE ( <a href="#">see page 499</a> ) = Allow lower case letters in file names. Default value is BS_FALSE ( <a href="#">see page 335</a> ).
◆	ISOAllowManyDirectories ( <a href="#">see page 61</a> )	BS_TRUE ( <a href="#">see page 499</a> ) = Allow path depth of more than 8 directories. Default value is BS_FALSE ( <a href="#">see page 335</a> ).
◆	ISOApplicationIdentifier ( <a href="#">see page 61</a> )	The Volume Application Identifier field of the ISO/Joliet image(128 characters maximum). For detailed information see: ISO 9660.
◆	ISOBiblioidentifier ( <a href="#">see page 61</a> )	The Volume Bibliography File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.
◆	ISOCopyrightFileIdentifier ( <a href="#">see page 61</a> )	Volume Copyright File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.
◆	ISOCreationDateTime ( <a href="#">see page 61</a> )	The creation date of the media according the ISO specification. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
◆	ISODataPreparerIdentifier ( <a href="#">see page 61</a> )	The Volume Data Preparer Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆	ISOEffectiveDateTime ( <a href="#">see page 62</a> )	The effective date of the media according the ISO specification. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
◆	ISOExpirationDateTime ( <a href="#">see page 62</a> )	The expiration date of the media according the ISO specification. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
◆	ISOLevel ( <a href="#">see page 62</a> )	This option will set the ISO Level for ISO9660. Possible values are: BS_ISO_LEVEL_1 ( <a href="#">see page 360</a> ), BS_ISO_LEVEL_2 ( <a href="#">see page 360</a> ), BS_ISO_LEVEL_3 ( <a href="#">see page 361</a> ), BS_ISO_LEVEL_ROMEO ( <a href="#">see page 361</a> )

◆	ISOModificationDateTime ( <a href="#">see page 62</a> )	The modification date of the media according the ISO specification. Use a SFileDateTime ( <a href="#">see page 38</a> ) structure as value.
◆	ISOPublisherIdentifier ( <a href="#">see page 62</a> )	The Volume Publisher Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆	ISOSetIdentifier ( <a href="#">see page 62</a> )	The Volume Set Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆	ISOSystemIdentifier ( <a href="#">see page 63</a> )	The Volume System Identifier field of the ISO/Joliet image (31 characters maximum). For detailed information see: ISO 9660.
◆	ISOUseCreationDateTime ( <a href="#">see page 63</a> )	Switch to activate the ISOCreationDateTime ( <a href="#">see page 61</a> ) in the ISO structure. If set to BS_TRUE ( <a href="#">see page 499</a> ) the SDK will use the value setted by ISOCreationDateTime ( <a href="#">see page 61</a> ). Default BS_FALSE ( <a href="#">see page 335</a> ), will use current date.
◆	ISOUseEffectiveDateTime ( <a href="#">see page 63</a> )	Switch to activate the ISOEffectiveDateTime ( <a href="#">see page 62</a> ) in the ISO structure. If set to BS_TRUE ( <a href="#">see page 499</a> ) the SDK will use the value setted by ISOEffectiveDateTime ( <a href="#">see page 62</a> ). Default BS_FALSE ( <a href="#">see page 335</a> ), will use current date.
◆	ISOUseExpirationDateTime ( <a href="#">see page 63</a> )	Switch to activate the ISOExpirationDateTime ( <a href="#">see page 62</a> ) in the ISO structure. If set to BS_TRUE ( <a href="#">see page 499</a> ) the SDK will use the value setted by ISOExpirationDateTime ( <a href="#">see page 62</a> ). Default BS_FALSE ( <a href="#">see page 335</a> ), will use current date.
◆	ISOUseModificationDateTime ( <a href="#">see page 63</a> )	Switch to activate the ISOModificationDateTime ( <a href="#">see page 62</a> ) in the ISO structure. If set to BS_TRUE ( <a href="#">see page 499</a> ) the SDK will use the value setted by ISOModificationDateTime ( <a href="#">see page 62</a> ). Default BS_FALSE ( <a href="#">see page 335</a> ), will use current date.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 64</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SISOInfoEx.
◆	SISOInfoEx ( <a href="#">see page 58</a> )	This is SISOInfoEx, a member of class SISOInfoEx.

## 1.1.21.2 SISOInfoEx::SISOInfoEx Constructor

**C++**

```
SISOInfoEx( ) ;
```

**Description**

This is SISOInfoEx, a member of class SISOInfoEx.

## 1.1.21.3 SISOInfoEx Data Members

The data members of the SISOInfoEx class are listed here.

**Public Data Members**

	Name	Description
◆	ISOAbstractFileIdentifier ( <a href="#">see page 60</a> )	The Volume Abstract File Identifier field of the ISO/Joliet image (35 characters maximum). For detailed information see: ISO 9660.
◆	ISOAddSuffix ( <a href="#">see page 60</a> )	Get/Set the bool value to enable ISO version number extension (1). The ISO 9660 specification requires that each filename include a ;x version number as the suffix. A full filename in ISO 9660 has three parts: <name>.<extension>;<version> Presumably the version part was designed to allow multiple versions of the same file to coexist.

◆ ISOAllowLongISO9660Names (see page 60)	If enabled (BS_TRUE (see page 499)), allows long file and directory names up to 207 ASCII characters in ISO9660 file system. If options is disabled (BS_FALSE (see page 335)), the ISO Level restrictions on file name length will be used.
◆ ISOAllowLongJolietNames (see page 60)	If enabled (BS_TRUE (see page 499)), allows long file and directory names up to 103 UCS-2 characters in Joliet extension to ISO9660 file system. If options is disabled (BS_FALSE (see page 335)), the max file name length will be limited by 64 UCS-2 characters.
◆ ISOAllowLowercaseNames (see page 60)	BS_TRUE (see page 499) = Allow lower case letters in file names. Default value is BS_FALSE (see page 335).
◆ ISOAllowManyDirectories (see page 61)	BS_TRUE (see page 499) = Allow path dept of more than 8 directories. Default value is BS_FALSE (see page 335).
◆ ISOApplicationIdentifier (see page 61)	The Volume Application Identifier field of the ISO/Joliet image(128 characters maximum). For detailed information see: ISO 9660.
◆ ISOBiblioidentifier (see page 61)	The Volume Bibliography File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.
◆ ISOCopyrightFileIdentifier (see page 61)	Volume Copyright File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.
◆ ISOCreationDateTime (see page 61)	The creation date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.
◆ ISODataPreparerIdentifier (see page 61)	The Volume Data Preparer Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆ ISOEffectiveDateTime (see page 62)	The effective date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.
◆ ISOExpirationDateTime (see page 62)	The expiration date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.
◆ ISOLevel (see page 62)	This option will set the ISO Level for ISO9660. Possible values are: BS_ISO_LEVEL_1 (see page 360), BS_ISO_LEVEL_2 (see page 360), BS_ISO_LEVEL_3 (see page 361), BS_ISO_LEVEL_ROMEO (see page 361)
◆ ISOModificationDateTime (see page 62)	The modification date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.
◆ ISOPublisherIdentifier (see page 62)	The Volume Publisher Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆ ISOSetIdentifier (see page 62)	The Volume Set Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.
◆ ISOSystemIdentifier (see page 63)	The Volume System Identifier field of the ISO/Joliet image (31 characters maximum). For detailed information see: ISO 9660.
◆ ISOUseCreationDateTime (see page 63)	Switch to activate the ISOCreationDateTime (see page 61) in the ISO structure. If set to BS_TRUE (see page 499) the SDK will use the value setted by ISOCreationDateTime (see page 61). Default BS_FALSE (see page 335), will use current date.
◆ ISOUseEffectiveDateTime (see page 63)	Switch to activate the ISOEffectiveDateTime (see page 62) in the ISO structure. If set to BS_TRUE (see page 499) the SDK will use the value setted by ISOEffectiveDateTime (see page 62). Default BS_FALSE (see page 335), will use current date.
◆ ISOUseExpirationDateTime (see page 63)	Switch to activate the ISOExpirationDateTime (see page 62) in the ISO structure. If set to BS_TRUE (see page 499) the SDK will use the value setted by ISOExpirationDateTime (see page 62). Default BS_FALSE (see page 335), will use current date.
◆ ISOUseModificationDateTime (see page 63)	Switch to activate the ISOModificationDateTime (see page 62) in the ISO structure. If set to BS_TRUE (see page 499) the SDK will use the value setted by ISOModificationDateTime (see page 62). Default BS_FALSE (see page 335), will use current date.

### 1.1.21.3.1 SISOInfoEx::ISOAbstractFileIdentifier Data Member

C++

```
TCHAR ISOAbstractFileIdentifier[ 36 ];
```

#### Description

The Volume Abstract File Identifier field of the ISO/Joliet image (35 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.2 SISOInfoEx::ISOAddSuffix Data Member

C++

```
BS_BOOL ISOAddSuffix;
```

#### Description

Get/Set the bool value to enable ISO version number extension (;1).

The ISO 9660 specification requires that each filename include a ;x version number as the suffix. A full filename in ISO 9660 has three parts:

<name>.<extension>;<version>

Presumably the version part was designed to allow multiple versions of the same file to coexist.

### 1.1.21.3.3 SISOInfoEx::ISOAllowLongISO9660Names Data Member

C++

```
BS_BOOL ISOAllowLongISO9660Names;
```

#### Description

If enabled (BS\_TRUE (see page 499)), allows long file and directory names up to 207 ASCII characters in ISO9660 file system. If options is disabled (BS\_FALSE (see page 335)), the ISO Level restrictions on file name length will be used.

### 1.1.21.3.4 SISOInfoEx::ISOAllowLongJolietNames Data Member

C++

```
BS_BOOL ISOAllowLongJolietNames;
```

#### Description

If enabled (BS\_TRUE (see page 499)), allows long file and directory names up to 103 UCS-2 characters in Joliet extension to ISO9660 file system. If options is disabled (BS\_FALSE (see page 335)), the max file name length will be limited by 64 UCS-2 characters.

### 1.1.21.3.5 SISOInfoEx::ISOAllowLowercaseNames Data Member

C++

```
BS_BOOL ISOAllowLowercaseNames;
```

#### Description

BS\_TRUE (see page 499) = Allow lower case letters in file names. Default value is BS\_FALSE (see page 335).

### 1.1.21.3.6 SISOInfoEx::ISOAllowManyDirectories Data Member

C++

```
BS_BOOL ISOAllowManyDirectories;
```

Description

BS\_TRUE (see page 499) = Allow path dept of more than 8 directories. Default value is BS\_FALSE (see page 335).

### 1.1.21.3.7 SISOInfoEx::ISOApplicationIdentifier Data Member

C++

```
TCHAR ISOApplicationIdentifier[128];
```

Description

The Volume Application Identifier field of the ISO/Joliet image(128 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.8 SISOInfoEx::ISOBiblioidentifier Data Member

C++

```
TCHAR ISOBiblioidentifier[36];
```

Description

The Volume Bibliography File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.9 SISOInfoEx::ISOCopyrightFileIdentifier Data Member

C++

```
TCHAR ISOCopyrightFileIdentifier[36];
```

Description

Volume Copyright File Identifier field of the ISO/Joliet image (36 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.10 SISOInfoEx::ISOCreationDateTime Data Member

C++

```
SFileDateTime ISOCreationDateTime;
```

Description

The creation date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.

### 1.1.21.3.11 SISOInfoEx::ISODataPreparerIdentifier Data Member

C++

```
TCHAR ISODataPreparerIdentifier[128];
```

Description

The Volume Data Preparer Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.12 SISOInfoEx::ISOEffectiveDateTime Data Member

C++

```
SFileDateTime ISOEffectiveDateTime;
```

**Description**

The effective date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.

### 1.1.21.3.13 SISOInfoEx::ISOExpirationDateTime Data Member

C++

```
SFileDateTime ISOExpirationDateTime;
```

**Description**

The expiration date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.

### 1.1.21.3.14 SISOInfoEx::ISOLevel Data Member

C++

```
int32 ISOLevel;
```

**Description**

This option will set the ISO Level for ISO9660. Possible values are:

BS\_ISO\_LEVEL\_1 (see page 360),

BS\_ISO\_LEVEL\_2 (see page 360),

BS\_ISO\_LEVEL\_3 (see page 361),

BS\_ISO\_LEVEL\_ROMEO (see page 361)

### 1.1.21.3.15 SISOInfoEx::ISOModificationDateTime Data Member

C++

```
SFileDateTime ISOModificationDateTime;
```

**Description**

The modification date of the media according the ISO specification. Use a SFileDateTime (see page 38) structure as value.

### 1.1.21.3.16 SISOInfoEx::ISOPublisherIdentifier Data Member

C++

```
TCHAR ISOPublisherIdentifier[128];
```

**Description**

The Volume Publisher Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.

### 1.1.21.3.17 SISOInfoEx::ISOSetIdentifier Data Member

C++

```
TCHAR ISOSetIdentifier[128];
```

### Description

The Volume Set Identifier field of the ISO/Joliet image (128 characters maximum). For detailed information see: ISO 9660.

## 1.1.21.3.18 SISOInfoEx::ISOSystemIdentifier Data Member

### C++

```
TCHAR ISOSystemIdentifier[31];
```

### Description

The Volume System Identifier field of the ISO/Joliet image (31 characters maximum). For detailed information see: ISO 9660.

## 1.1.21.3.19 SISOInfoEx::ISOUseCreationDateTime Data Member

### C++

```
BS_BOOL ISOUseCreationDateTime;
```

### Description

Switch to activate the ISOCreationDateTime (see page 61) in the ISO structure. If set to BS\_TRUE (see page 499) the SDK will use the value setted by ISOCreationDateTime (see page 61). Default BS\_FALSE (see page 335), will use current date.

## 1.1.21.3.20 SISOInfoEx::ISOUseEffectiveDateTime Data Member

### C++

```
BS_BOOL ISOUseEffectiveDateTime;
```

### Description

Switch to activate the ISOEffectiveDateTime (see page 62) in the ISO structure. If set to BS\_TRUE (see page 499) the SDK will use the value setted by ISOEffectiveDateTime (see page 62). Default BS\_FALSE (see page 335), will use current date.

## 1.1.21.3.21 SISOInfoEx::ISOUseExpirationDateTime Data Member

### C++

```
BS_BOOL ISOUseExpirationDateTime;
```

### Description

Switch to activate the ISOExpirationDateTime (see page 62) in the ISO structure. If set to BS\_TRUE (see page 499) the SDK will use the value setted by ISOExpirationDateTime (see page 62). Default BS\_FALSE (see page 335), will use current date.

## 1.1.21.3.22 SISOInfoEx::ISOUseModificationDateTime Data Member

### C++

```
BS_BOOL ISOUseModificationDateTime;
```

### Description

Switch to activate the ISOModificationDateTime (see page 62) in the ISO structure. If set to BS\_TRUE (see page 499) the SDK will use the value setted by ISOModificationDateTime (see page 62). Default BS\_FALSE (see page 335), will use current date.

## 1.1.21.4 SISOInfoEx Methods

The methods of the SISOInfoEx class are listed here.

### Public Methods

	Name	Description
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 64</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SISOInfoEx.

### 1.1.21.4.1 SISOInfoEx::INSERT\_STRUCTURE\_PADDING Method

#### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - 6 * sizeof(BS_BOOL) - 4 * sizeof(SFileDateTime));
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SISOInfoEx.

## 1.1.22 SISOVolumeInfo Structure

#### C++

```
struct SISOVolumeInfo {
    TCHAR chVolumeLabel[128];
    int32 nVolumeDescriptorAddress;
    int32 nVolumeSize;
    int32 nRootAddress;
    int32 nPathTableAddress;
    int32 nPathTableSize;
    SFileDateTime tRootDateTime;
    struct SISOInfoEx sInfoEx;
};
```

#### File

IsoSDKDefinitions.h ([see page 517](#))

#### Description

IsoSDK sub class that contains the information about the ISO information.

### 1.1.22.1 SISOVolumeInfo Members

The following tables list the members exposed by SISOVolumeInfo.

#### Public Data Members

	Name	Description
❖	chVolumeLabel ( <a href="#">see page 65</a> )	The volume label name.
❖	nPathTableAddress ( <a href="#">see page 65</a> )	The address where the path table starts.
❖	nPathTableSize ( <a href="#">see page 66</a> )	The size of the path table.
❖	nRootAddress ( <a href="#">see page 66</a> )	The root address of the ISO file system.
❖	nVolumeDescriptorAddress ( <a href="#">see page 66</a> )	The address where the volume descriptor starts.
❖	nVolumeSize ( <a href="#">see page 66</a> )	The size of the disc.

	sInfoEx ( <a href="#">see page 66</a> )	A structure SISOInfoEx ( <a href="#">see page 56</a> ).
	tRootDateTime ( <a href="#">see page 66</a> )	The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

**Public Methods**

	Name	Description
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 67</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SISOVolumeInfo.
	SISOVolumeInfo ( <a href="#">see page 65</a> )	A structure that contains the information about the ISO information.

## 1.1.22.2 SISOVolumeInfo::SISOVolumeInfo Constructor

**C++**`SISOVolumeInfo();`**Description**

A structure that contains the information about the ISO information.

## 1.1.22.3 SISOVolumeInfo Data Members

The data members of the SISOVolumeInfo class are listed here.

**Public Data Members**

	Name	Description
	chVolumeLabel ( <a href="#">see page 65</a> )	The volume label name.
	nPathTableAddress ( <a href="#">see page 65</a> )	The address where the path table starts.
	nPathTableSize ( <a href="#">see page 66</a> )	The size of the path table.
	nRootAddress ( <a href="#">see page 66</a> )	The root address of the ISO file system.
	nVolumeDescriptorAddress ( <a href="#">see page 66</a> )	The address where the volume descriptor starts.
	nVolumeSize ( <a href="#">see page 66</a> )	The size of the disc.
	sInfoEx ( <a href="#">see page 66</a> )	A structure SISOInfoEx ( <a href="#">see page 56</a> ).
	tRootDateTime ( <a href="#">see page 66</a> )	The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

### 1.1.22.3.1 SISOVolumeInfo::chVolumeLabel Data Member

**C++**`TCHAR chVolumeLabel[128];`**Description**

The volume label name.

### 1.1.22.3.2 SISOVolumeInfo::nPathTableAddress Data Member

**C++**`int32 nPathTableAddress;`

**Description**

The address where the path table starts.

### 1.1.22.3.3 SISOVolumeInfo::nPathTableSize Data Member

**C++**

```
int32 nPathTableSize;
```

**Description**

The size of the path table.

### 1.1.22.3.4 SISOVolumeInfo::nRootAddress Data Member

**C++**

```
int32 nRootAddress;
```

**Description**

The root address of the ISO file system.

### 1.1.22.3.5 SISOVolumeInfo::nVolumeDescriptorAddress Data Member

**C++**

```
int32 nVolumeDescriptorAddress;
```

**Description**

The address where the volume descriptor starts.

### 1.1.22.3.6 SISOVolumeInfo::nVolumeSize Data Member

**C++**

```
int32 nVolumeSize;
```

**Description**

The size of the disc.

### 1.1.22.3.7 SISOVolumeInfo::sInfoEx Data Member

**C++**

```
struct SISOInfoEx sInfoEx;
```

**Description**

A structure SISOInfoEx (see page 56).

### 1.1.22.3.8 SISOVolumeInfo::tRootDateTime Data Member new

**C++**

```
SFileDateTime tRootDateTime;
```

**Description**

The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

## 1.1.22.4 SISOVolumeInfo Methods

The methods of the SISOVolumeInfo class are listed here.

### Public Methods

	Name	Description
✳	INSERT_STRUCTURE_PADDING ( <a href="#">see page 67</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SISOVolumeInfo.

### 1.1.22.4.1 SISOVolumeInfo::INSERT\_STRUCTURE\_PADDING Method

#### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA);
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SISOVolumeInfo.

## 1.1.23 SMediumInfo Structure

#### C++

```
struct SMediumInfo {
    TCHAR chMediumType[25];
    int32 nMediumStatus;
    double dMediumSize;
    double dMediumUsedSize;
    double dMediumFreeSize;
    int32 nFirstSession;
    int32 nLastSession;
    int32 nFirstTrack;
    int32 nLastTrack;
    int32 nLastSessionStatus;
    int16 wMediumTypeCode;
    int16 nExtendedMediumType;
    TCHAR chUPCEANCode[14];
    TCHAR chVendorID[20];
    float fMaxWriteSpeed;
};
```

#### File

IsoSDKDefinitions.h ([see page 517](#))

#### Description

IsoSDK sub class that contains the information about the medium.

## 1.1.23.1 SMediumInfo Members

The following tables list the members exposed by SMediumInfo.

### Public Data Members

	Name	Description
✳	chMediumType ( <a href="#">see page 71</a> )	Returns the medium type in clear text.
✳	chUPCEANCode ( <a href="#">see page 71</a> )	Returns the UPCEAN Code of the current disc.
✳	chVendorID ( <a href="#">see page 71</a> )	The vendor ID of the disc producer.

◆	dMediumFreeSize ( <a href="#">see page 72</a> )	The available space of the disc in bytes.
◆	dMediumSize ( <a href="#">see page 72</a> )	The current disc size in bytes.
◆	dMediumUsedSize ( <a href="#">see page 72</a> )	Already used space of the current disc in bytes.
◆	fMaxWriteSpeed ( <a href="#">see page 72</a> )	This is the max. write speed the drive reported. This is not supported by all drives.
◆	nExtendedMediumType ( <a href="#">see page 72</a> )	The type of the current disc. The following values will be returned: BS_EMT_CD_ROM ( <a href="#">see page 331</a> ) BS_EMT_CD_ROM_XA ( <a href="#">see page 331</a> ) BS_EMT_CD_AUDIO ( <a href="#">see page 330</a> ) BS_EMT_CD_MIXED_MODE ( <a href="#">see page 330</a> ) BS_EMT_CD_ENHANCED ( <a href="#">see page 330</a> ) BS_EMT_CD_MULTISESSION ( <a href="#">see page 330</a> ) BS_EMT_DVD ( <a href="#">see page 331</a> )
◆	nFirstSession ( <a href="#">see page 73</a> )	The number of the first session of the current disc.
◆	nFirstTrack ( <a href="#">see page 73</a> )	The number of the first track of the current disc.
◆	nLastSession ( <a href="#">see page 73</a> )	The number of the last session of the current disc.
◆	nLastSessionStatus ( <a href="#">see page 73</a> )	The status of the last session. The following values will be returned: BS_LS_EMPTY_SESSION ( <a href="#">see page 364</a> ), BS_LS_INCOMPLETE_SESSION ( <a href="#">see page 364</a> ), BS_LS_DAMAGED_SESSION ( <a href="#">see page 363</a> ), BS_LS_COMPLETE_SESSION ( <a href="#">see page 363</a> )
◆	nLastTrack ( <a href="#">see page 73</a> )	The number of the last track of the current disc.
◆	nMediumStatus ( <a href="#">see page 73</a> )	The status of the current disc. The following values will be returned: BS_MS_EMPTY_DISK ( <a href="#">see page 365</a> ), BS_MS_INCOMPLETE_DISK ( <a href="#">see page 366</a> ), BS_MS_COMPLETE_DISK ( <a href="#">see page 365</a> ), BS_MS_OTHER ( <a href="#">see page 366</a> )

 wMediumTypeCode ( <a href="#">see page 74</a> )	The type of the current disc. The following values will be returned: BS_UNKNOWN ( <a href="#">see page 503</a> ) BS_CD_ROM ( <a href="#">see page 317</a> ) BS_CD_R ( <a href="#">see page 317</a> ) BS_CD_RW ( <a href="#">see page 318</a> ) BS_DVD_ROM ( <a href="#">see page 328</a> ) BS_DVD_R ( <a href="#">see page 327</a> ) BS_DVD_RAM ( <a href="#">see page 327</a> ) BS_DVD_RW_RO ( <a href="#">see page 328</a> ) BS_DVD_RW ( <a href="#">see page 328</a> ) BS_DVD_RW_SR ( <a href="#">see page 329</a> ) BS_DVD_PLUSRW ( <a href="#">see page 327</a> ) BS_DVD_PLUSR ( <a href="#">see page 327</a> ) BS_DD_CD_ROM ( <a href="#">see page 324</a> ) BS_DD_CD_R ( <a href="#">see page 324</a> ) BS_DD_CD_RW ( <a href="#">see page 324</a> ) BS_DVD_RDL_PLUS ( <a href="#">see page 328</a> ) BS_DVD_RWDL_PLUS ( <a href="#">see page 329</a> ) BS_DVD_MRDL ( <a href="#">see page 326</a> ) BS_BLURAY_R ( <a href="#">see page 311</a> ) BS_BLURAY_RE ( <a href="#">see page 312</a> ) BS_BLURAY_ROM ( <a href="#">see page 312</a> ) BS_BLURAY_R_RRM ( <a href="#">see page 312</a> ) BS_HD_DVD_R ( <a href="#">see page 355</a> ) BS_HD_DVD_RW ( <a href="#">see page 356</a> ) BS_HD_DVD_ROM ( <a href="#">see page 355</a> ) BS_HD_DVD_RAM ( <a href="#">see page 355</a> ) BS_HD_DVD_R_DL ( <a href="#">see page 355</a> ) BS_HD_DVD_RW_DL ( <a href="#">see page 356</a> )
---	--

**Public Methods**

	<b>Name</b>	<b>Description</b>
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 75</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SMediumInfo.
	SMediumInfo ( <a href="#">see page 69</a> )	This is SMediumInfo, a member of class SMediumInfo.

**1.1.23.2 SMediumInfo::SMediumInfo Constructor****C++**

```
SMediumInfo();
```

**Description**

This is SMediumInfo, a member of class SMediumInfo.

**1.1.23.3 SMediumInfo Data Members**

The data members of the SMediumInfo class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
	chMediumType ( <a href="#">see page 71</a> )	Returns the medium type in clear text.
	chUPCEANCode ( <a href="#">see page 71</a> )	Returns the UPCEAN Code of the current disc.
	chVendorID ( <a href="#">see page 71</a> )	The vendor ID of the disc producer.

◆	dMediumFreeSize ( <a href="#">see page 72</a> )	The available space of the disc in bytes.
◆	dMediumSize ( <a href="#">see page 72</a> )	The current disc size in bytes.
◆	dMediumUsedSize ( <a href="#">see page 72</a> )	Already used space of the current disc in bytes.
◆	fMaxWriteSpeed ( <a href="#">see page 72</a> )	This is the max. write speed the drive reported. This is not supported by all drives.
◆	nExtendedMediumType ( <a href="#">see page 72</a> )	The type of the current disc. The following values will be returned: BS_EMT_CD_ROM ( <a href="#">see page 331</a> ) BS_EMT_CD_ROM_XA ( <a href="#">see page 331</a> ) BS_EMT_CD_AUDIO ( <a href="#">see page 330</a> ) BS_EMT_CD_MIXED_MODE ( <a href="#">see page 330</a> ) BS_EMT_CD_ENHANCED ( <a href="#">see page 330</a> ) BS_EMT_CD_MULTISESSION ( <a href="#">see page 330</a> ) BS_EMT_DVD ( <a href="#">see page 331</a> )
◆	nFirstSession ( <a href="#">see page 73</a> )	The number of the first session of the current disc.
◆	nFirstTrack ( <a href="#">see page 73</a> )	The number of the first track of the current disc.
◆	nLastSession ( <a href="#">see page 73</a> )	The number of the last session of the current disc.
◆	nLastSessionStatus ( <a href="#">see page 73</a> )	The status of the last session. The following values will be returned: BS_LS_EMPTY_SESSION ( <a href="#">see page 364</a> ), BS_LS_INCOMPLETE_SESSION ( <a href="#">see page 364</a> ), BS_LS_DAMAGED_SESSION ( <a href="#">see page 363</a> ), BS_LS_COMPLETE_SESSION ( <a href="#">see page 363</a> )
◆	nLastTrack ( <a href="#">see page 73</a> )	The number of the last track of the current disc.
◆	nMediumStatus ( <a href="#">see page 73</a> )	The status of the current disc. The following values will be returned: BS_MS_EMPTY_DISK ( <a href="#">see page 365</a> ), BS_MS_INCOMPLETE_DISK ( <a href="#">see page 366</a> ), BS_MS_COMPLETE_DISK ( <a href="#">see page 365</a> ), BS_MS_OTHER ( <a href="#">see page 366</a> )

<span style="color: #0070C0;">◆</span>	<b>wMediumTypeCode</b> ( [ see page 74 ) The type of the current disc. The following values will be returned: BS_UNKNOWN ( [ see page 503 ) BS_CD_ROM ( [ see page 317 ) BS_CD_R ( [ see page 317 ) BS_CD_RW ( [ see page 318 ) BS_DVD_ROM ( [ see page 328 ) BS_DVD_R ( [ see page 327 ) BS_DVD_RAM ( [ see page 327 ) BS_DVD_RW_RO ( [ see page 328 ) BS_DVD_RW ( [ see page 328 ) BS_DVD_RW_SR ( [ see page 329 ) BS_DVD_PLUSRW ( [ see page 327 ) BS_DVD_PLUSR ( [ see page 327 ) BS_DD_CD_ROM ( [ see page 324 ) BS_DD_CD_R ( [ see page 324 ) BS_DD_CD_RW ( [ see page 324 ) BS_DVD_RDL_PLUS ( [ see page 328 ) BS_DVD_RWDL_PLUS ( [ see page 329 ) BS_DVD_MRDL ( [ see page 326 ) BS_BLURAY_R ( [ see page 311 ) BS_BLURAY_RE ( [ see page 312 ) BS_BLURAY_ROM ( [ see page 312 ) BS_BLURAY_R_RRM ( [ see page 312 ) BS_HD_DVD_R ( [ see page 355 ) BS_HD_DVD_RW ( [ see page 356 ) BS_HD_DVD_ROM ( [ see page 355 ) BS_HD_DVD_RAM ( [ see page 355 ) BS_HD_DVD_R_DL ( [ see page 355 ) BS_HD_DVD_RW_DL ( [ see page 356 )
--	--

### 1.1.23.3.1 SMediumInfo::chMediumType Data Member

**C++**

```
TCHAR chMediumType[ 25 ];
```

**Description**

Returns the medium type in clear text.

### 1.1.23.3.2 SMediumInfo::chUPCEANCode Data Member

**C++**

```
TCHAR chUPCEANCode[ 14 ];
```

**Description**

Returns the UPCEAN Code of the current disc.

### 1.1.23.3.3 SMediumInfo::chVendorID Data Member

**C++**

```
TCHAR chVendorID[ 20 ];
```

**Description**

The vendor ID of the disc producer.

### 1.1.23.3.4 SMediumInfo::dMediumFreeSize Data Member

C++

```
double dMediumFreeSize;
```

Description

The available space of the disc in bytes.

### 1.1.23.3.5 SMediumInfo::dMediumSize Data Member

C++

```
double dMediumSize;
```

Description

The current disc size in bytes.

### 1.1.23.3.6 SMediumInfo::dMediumUsedSize Data Member

C++

```
double dMediumUsedSize;
```

Description

Already used space of the current disc in bytes.

### 1.1.23.3.7 SMediumInfo::fMaxWriteSpeed Data Member

C++

```
float fMaxWriteSpeed;
```

Description

This is the max. write speed the drive reported. This is not supported by all drives.

### 1.1.23.3.8 SMediumInfo::nExtendedMediumType Data Member

C++

```
int16 nExtendedMediumType;
```

Description

The type of the current disc. The following values will be returned:

BS\_EMT\_CD\_ROM ([see page 331](#))

BS\_EMT\_CD\_ROM\_XA ([see page 331](#))

BS\_EMT\_CD\_AUDIO ([see page 330](#))

BS\_EMT\_CD\_MIXED\_MODE ([see page 330](#))

BS\_EMT\_CD\_ENHANCED ([see page 330](#))

BS\_EMT\_CD\_MULTISESSION ([see page 330](#))

BS\_EMT\_DVD ([see page 331](#))

### 1.1.23.3.9 SMediumInfo::nFirstSession Data Member

C++

```
int32 nFirstSession;
```

**Description**

The number of the first session of the current disc.

### 1.1.23.3.10 SMediumInfo::nFirstTrack Data Member

C++

```
int32 nFirstTrack;
```

**Description**

The number of the first track of the current disc.

### 1.1.23.3.11 SMediumInfo::nLastSession Data Member

C++

```
int32 nLastSession;
```

**Description**

The number of the last session of the current disc.

### 1.1.23.3.12 SMediumInfo::nLastSessionStatus Data Member

C++

```
int32 nLastSessionStatus;
```

**Description**

The status of the last session. The following values will be returned:

BS\_LS\_EMPTY\_SESSION (see page 364),  
BS\_LS\_INCOMPLETE\_SESSION (see page 364),  
BS\_LS\_DAMAGED\_SESSION (see page 363),  
BS\_LS\_COMPLETE\_SESSION (see page 363)

### 1.1.23.3.13 SMediumInfo::nLastTrack Data Member

C++

```
int32 nLastTrack;
```

**Description**

The number of the last track of the current disc.

### 1.1.23.3.14 SMediumInfo::nMediumStatus Data Member

C++

```
int32 nMediumStatus;
```

**Description**

The status of the current disc. The following values will be returned:

BS\_MS\_EMPTY\_DISK (see page 365),  
BS\_MS\_INCOMPLETE\_DISK (see page 366),  
BS\_MS\_COMPLETE\_DISK (see page 365),  
BS\_MS\_OTHER (see page 366)

### 1.1.23.3.15 SMediumInfo::wMediumTypeCode Data Member

#### C++

```
int16 wMediumTypeCode;
```

#### Description

The type of the current disc. The following values will be returned:

BS\_UNKNOWN (see page 503)  
BS\_CD\_ROM (see page 317)  
BS\_CD\_R (see page 317)  
BS\_CD\_RW (see page 318)  
BS\_DVD\_ROM (see page 328)  
BS\_DVD\_R (see page 327)  
BS\_DVD\_RAM (see page 327)  
BS\_DVD\_RW\_RO (see page 328)  
BS\_DVD\_RW (see page 328)  
BS\_DVD\_RW\_SR (see page 329)  
BS\_DVD\_PLUSRW (see page 327)  
BS\_DVD\_PLUSR (see page 327)  
BS\_DD\_CD\_ROM (see page 324)  
BS\_DD\_CD\_R (see page 324)  
BS\_DD\_CD\_RW (see page 324)  
BS\_DVD\_RDL\_PLUS (see page 328)  
BS\_DVD\_RWDL\_PLUS (see page 329)  
BS\_DVD\_MRDL (see page 326)  
BS\_BLURAY\_R (see page 311)  
BS\_BLURAY\_RE (see page 312)  
BS\_BLURAY\_ROM (see page 312)  
BS\_BLURAY\_R\_RRM (see page 312)  
BS\_HD\_DVD\_R (see page 355)  
BS\_HD\_DVD\_RW (see page 356)  
BS\_HD\_DVD\_ROM (see page 355)  
BS\_HD\_DVD\_RAM (see page 355)  
BS\_HD\_DVD\_R\_DL (see page 355)  
BS\_HD\_DVD\_RW\_DL (see page 356)

## 1.1.23.4 SMediumInfo Methods

The methods of the SMediumInfo class are listed here.

### Public Methods

	Name	Description
☞	INSERT_STRUCTURE_PADDING (☞ see page 75)	This is INSERT_STRUCTURE_PADDING, a member of class SMediumInfo.

### 1.1.23.4.1 SMediumInfo::INSERT\_STRUCTURE\_PADDING Method

#### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_HUGE - sizeof(TCHAR[ 20 ]) - sizeof(float));
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SMediumInfo.

## 1.1.24 SOptions Structure

#### C++

```
struct SOptions {
    TCHAR chVolumeLabel[128];
    int32 nWriteMethod;
    BS_BOOL bJolietFileSystem;
    BS_BOOL bBootable;
    TCHAR chBootImage[_MAX_PATH];
    BS_BOOL bFinalize;
    BS_BOOL bTestBurn;
    BS_BOOL bPerformOPC;
    BS_BOOL bVerifyAfterBurn;
    int32 nCacheSize;
    BS_BOOL bUnderrunProtection;
    BS_BOOL bEjectAfterBurn;
    BS_BOOL bAutoErase;
    BS_BOOL bRockRidgeFileSystem;
    int32 nCopies;
    BS_BOOL bPadDataTracks;
};
```

#### File

IsoSDKDefinitions.h (☞ see page 517)

#### Description

IsoSDK sub class that is used to get and set the parameters for the current project.

1 = BS_BURNER_DATA (☞ see page 315)	2 = BS_BURNER_UDFDVD (☞ see page 316)	3 = BS_BURNER_ISOUDF (☞ see page 315)
4 = BS_BURNER_BLURAY (☞ see page 314)	5 = BS_BURNER_AUDIO (☞ see page 314)	6 = BS_BURNER_MIXEDMODE (☞ see page 315)
7 = BS_BURNER_VIDEODVD (☞ see page 316)	8 = BS_BURNER_VCD (☞ see page 316)	9 = BS_BURNER_SVCD (☞ see page 316)
10 = BS_BURNER_CUE (☞ see page 314)	11 = BS_BURNER_RAW (☞ see page 315)	

	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>
Label	x	x	x	x		x	x	x	x		
Write Method *1	x	x	x		x	x	x	x	x		
Joliet	x		x			x					
Boot	x		x								
Finalize	x	x	x	x							
Test Burning *1	x	x	x	x	x	x	x	x	x	x	x
OPC	x	x	x	x	x	x	x	x	x	x	x
Verify	x	x	x	x		x *2	x				
Eject	x	x	x	x	x	x	x	x	x	x	x
UnderrunProtection	x	x	x	x	x	x	x	x	x	x	x
UDF		x	x	x							
IsoEx	x		x			x	x				
Compression	x										
Encryption	x										
Pad Data Tracks	x	x	x	x		x					

\*1 = Only valid for disc type CD

\*2 = Only the ISO part will get verified

## 1.1.24.1 SOptions Members

The following tables list the members exposed by SOptions.

### Public Data Members

	<b>Name</b>	<b>Description</b>
◆	bAutoErase (see page 79)	This argument sets the autoerase function for CD/DVD/BluRay Rewriteable. If this option is set to BS_TRUE (see page 499) the FoxBurner SDK will erase a RW before the new burnjob started. Please note that this option use the "Quick erase" function.
◆	bBootable (see page 79)	If you want to make a bootable device, pass BS_TRUE (see page 499) to activate it and pass BS_FALSE (see page 335) to deactivate it. This option is only available for data projects.
◆	bEjectAfterBurn (see page 79)	This argument determines whether the medium will be ejected after the burning process. BS_TRUE (see page 499) ejects the medium, and BS_FALSE (see page 335) prevents this.
◆	bFinalize (see page 79)	This property states whether the medium should be finalized. BS_TRUE (see page 499) finalizes the medium, whereas BS_FALSE (see page 335) burns a multi-session medium.
◆	bJolietFileSystem (see page 79)	If you want to use the Joliet file system, pass BS_TRUE (see page 499) to activate it and pass BS_FALSE (see page 335) to deactivate it. This option is only available for data projects.
◆	bPadDataTracks (see page 80)	This option will pad the data track of a disc to the min. size of 4 seconds. This is important for Data, Multisession and Mixed Mode disc if your work with very small sizes.

◆	bPerformOPC ( <a href="#">see page 80</a> )	DVD writing devices perform a test write and read in an area inside of the lead-in, in order to determine the best laser power for recording. This allows the device to adjust to each disc, which may vary slightly from different manufacturers, or for other reasons.  BS_TRUE ( <a href="#">see page 499</a> )/BS_FALSE ( <a href="#">see page 335</a> ) whether you want to perform OPC or not. Pass true to activate it and pass false to de-activate it.
◆	bRockRidgeFileSystem ( <a href="#">see page 80</a> )	This argument disable or enable the RockRidge support. Default value is BS_FALSE ( <a href="#">see page 335</a> ). This option is supported in following projects: Data, ISO/UDF, Mixed Mode.
◆	bTestBurn ( <a href="#">see page 80</a> )	Get/Set the simulation option. The value "BS_TRUE ( <a href="#">see page 499</a> )" activates the burning simulation and false skips the simulation. The IsoSDK will make a dry run with this option. To use the final write to the disc you have to set this value to false.
◆	bUnderrunProtection ( <a href="#">see page 80</a> )	You can activate the buffer-underrun-protection by setting the value to BS_TRUE ( <a href="#">see page 499</a> ). BS_FALSE ( <a href="#">see page 335</a> ) de-activates the buffer protection.
◆	bVerifyAfterBurn ( <a href="#">see page 81</a> )	This argument enables or disables the verify after burning. This option is only available for Data CD/DVD/BD ISO and UDF projects.
◆	chBootImage ( <a href="#">see page 81</a> )	File name of the image (with path information)
◆	chVolumeLabel ( <a href="#">see page 81</a> )	This argument sets the volume name of the medium (null terminated). This name will be displayed e.g. in the Windows Explorer. This string may neither contain high characters nor spaces.
◆	nCacheSize ( <a href="#">see page 81</a> )	Sets the size of the burning buffer to be used (in bytes; default: 4 * 1024 * 1024 [4 MB]) We recommend to use a range between 4 MB and 32 MB. 4 MB = CD 16 MB = DVD 32 MB = BD
◆	nCopies ( <a href="#">see page 81</a> )	This argument sets the number of copies to burn. Only available in GUI mode.
◆	nWriteMethod ( <a href="#">see page 81</a> )	This parameter sets the burning method. This option is only allowed for CD-R/CD-RW media. For DVD media it is ignored. Values are: BS_WM_TAO ( <a href="#">see page 507</a> ), BS_WM_DAO ( <a href="#">see page 506</a> ), BS_WM.DAO96 ( <a href="#">see page 506</a> )

## Public Methods

	Name	Description
◆	Empty ( <a href="#">see page 82</a> )	This is Empty, a member of class SOptions.
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 82</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SOptions.
◆	SOptions ( <a href="#">see page 77</a> )	This is SOptions, a member of class SOptions.

## 1.1.24.2 SOptions::SOptions Constructor

### C++

```
SOptions();
```

### Description

This is SOptions, a member of class SOptions.

## 1.1.24.3 SOptions Data Members

The data members of the SOptions class are listed here.

### Public Data Members

	Name	Description
◆	bAutoErase ( <a href="#">see page 79</a> )	This argument sets the autoerase function for CD/DVD/BluRay ReWritable. If this option is set to BS_TRUE ( <a href="#">see page 499</a> ) the FoxBurner SDK will erase a RW before the new burnjob started. Please note that this option use the "Quick erase" function.
◆	bBootable ( <a href="#">see page 79</a> )	If you want to make a bootable device, pass BS_TRUE ( <a href="#">see page 499</a> ) to activate it and pass BS_FALSE ( <a href="#">see page 335</a> ) to deactivate it. This option is only available for data projects.
◆	bEjectAfterBurn ( <a href="#">see page 79</a> )	This argument determines whether the medium will be ejected after the burning process. BS_TRUE ( <a href="#">see page 499</a> ) ejects the medium, and BS_FALSE ( <a href="#">see page 335</a> ) prevents this.
◆	bFinalize ( <a href="#">see page 79</a> )	This property states whether the medium should be finalized. BS_TRUE ( <a href="#">see page 499</a> ) finalizes the medium, whereas BS_FALSE ( <a href="#">see page 335</a> ) burns a multi-session medium.
◆	bJolietFileSystem ( <a href="#">see page 79</a> )	If you want to use the Joliet file system, pass BS_TRUE ( <a href="#">see page 499</a> ) to activate it and pass BS_FALSE ( <a href="#">see page 335</a> ) to deactivate it. This option is only available for data projects.
◆	bPadDataTracks ( <a href="#">see page 80</a> )	This option will pad the data track of a disc to the min. size of 4 seconds. This is important for Data, Multisession and Mixed Mode disc if your work with very small sizes.
◆	bPerformOPC ( <a href="#">see page 80</a> )	DVD writing devices perform a test write and read in an area inside of the lead-in, in order to determine the best laser power for recording. This allows the device to adjust to each disc, which may vary slightly from different manufacturers, or for other reasons.  BS_TRUE ( <a href="#">see page 499</a> )/BS_FALSE ( <a href="#">see page 335</a> ) whether you want to perform OPC or not. Pass true to activate it and pass false to de-activate it.
◆	bRockRidgeFileSystem ( <a href="#">see page 80</a> )	This argument disable or enable the RockRidge support. Default value is BS_FALSE ( <a href="#">see page 335</a> ). This option is supported in following projects: Data, ISO/UDF, Mixed Mode.
◆	bTestBurn ( <a href="#">see page 80</a> )	Get/Set the simulation option. The value "BS_TRUE ( <a href="#">see page 499</a> )" activates the burning simulation and false skips the simulation.  The IsoSDK will make a dry run with this option. To use the final write to the disc you have to set this value to false.
◆	bUnderrunProtection ( <a href="#">see page 80</a> )	You can activate the buffer-underrun-protection by setting the value to BS_TRUE ( <a href="#">see page 499</a> ). BS_FALSE ( <a href="#">see page 335</a> ) de-activates the buffer protection.
◆	bVerifyAfterBurn ( <a href="#">see page 81</a> )	This argument enables or disables the verify after burning. This option is only available for Data CD/DVD/BD ISO and UDF projects.
◆	chBootImage ( <a href="#">see page 81</a> )	File name of the image (with path information)
◆	chVolumeLabel ( <a href="#">see page 81</a> )	This argument sets the volume name of the medium (null terminated). This name will be displayed e.g. in the Windows Explorer. This string may neither contain high characters nor spaces.
◆	nCacheSize ( <a href="#">see page 81</a> )	Sets the size of the burning buffer to be used (in bytes; default: 4 * 1024 * 1024 [4 MB])  We recommend to use a range between 4 MB and 32 MB. 4 MB = CD 16 MB = DVD 32 MB = BD

◆	nCopies ( <a href="#">see page 81</a> )	This argument sets the number of copies to burn. Only available in GUI mode.
◆	nWriteMethod ( <a href="#">see page 81</a> )	This parameter sets the burning method. This option is only allowed for CD-R/CD-RW media. For DVD media it is ignored. Values are: BS_WM_TAO ( <a href="#">see page 507</a> ), BS_WM_DAO ( <a href="#">see page 506</a> ), BS_WM.DAO96 ( <a href="#">see page 506</a> )

### 1.1.24.3.1 SOptions::bAutoErase Data Member

C++

```
BS_BOOL bAutoErase;
```

#### Description

This argument sets the autoerase function for CD/DVD/BluRay ReWritable. If this option is set to BS\_TRUE ([see page 499](#)) the FoxBurner SDK will erase a RW before the new burnjob started. Please note that this option use the "Quick erase" function.

### 1.1.24.3.2 SOptions::bBootable Data Member

C++

```
BS_BOOL bBootable;
```

#### Description

If you want to make a bootable device, pass BS\_TRUE ([see page 499](#)) to activate it and pass BS\_FALSE ([see page 335](#)) to deactivate it. This option is only available for data projects.

### 1.1.24.3.3 SOptions::bEjectAfterBurn Data Member

C++

```
BS_BOOL bEjectAfterBurn;
```

#### Description

This argument determines whether the medium will be ejected after the burning process. BS\_TRUE ([see page 499](#)) ejects the medium, and BS\_FALSE ([see page 335](#)) prevents this.

### 1.1.24.3.4 SOptions::bFinalize Data Member

C++

```
BS_BOOL bFinalize;
```

#### Description

This property states whether the medium should be finalized. BS\_TRUE ([see page 499](#)) finalizes the medium, whereas BS\_FALSE ([see page 335](#)) burns a multi-session medium.

### 1.1.24.3.5 SOptions::bJolietFileSystem Data Member

C++

```
BS_BOOL bJolietFileSystem;
```

#### Description

If you want to use the Joliet file system, pass BS\_TRUE ([see page 499](#)) to activate it and pass BS\_FALSE ([see page 335](#)) to deactivate it. This option is only available for data projects.

### 1.1.24.3.6 SOptions::bPadDataTracks Data Member

C++

```
BS_BOOL bPadDataTracks;
```

**Description**

This option will pad the data track of a disc to the min. size of 4 seconds. This is important for Data, Multisession and Mixed Mode disc if your work with very small sizes.

### 1.1.24.3.7 SOptions::bPerformOPC Data Member

C++

```
BS_BOOL bPerformOPC;
```

**Description**

DVD writing devices perform a test write and read in an area inside of the lead-in, in order to determine the best laser power for recording. This allows the device to adjust to each disc, which may vary slightly from different manufacturers, or for other reasons.

BS\_TRUE (see page 499)/BS\_FALSE (see page 335) whether you want to perform OPC or not. Pass true to activate it and pass false to de-activate it.

### 1.1.24.3.8 SOptions::bRockRidgeFileSystem Data Member

C++

```
BS_BOOL bRockRidgeFileSystem;
```

**Description**

This argument disable or enable the RockRidge support. Default value is BS\_FALSE (see page 335). This option is supported in following projects: Data, ISO/UDF, Mixed Mode.

### 1.1.24.3.9 SOptions::bTestBurn Data Member

C++

```
BS_BOOL bTestBurn;
```

**Description**

Get/Set the simulation option. The value "BS\_TRUE (see page 499)" activates the burning simulation and false skips the simulation.

The IsoSDK will make a dry run with this option. To use the final write to the disc you have to set this value to false.

### 1.1.24.3.10 SOptions::bUnderrunProtection Data Member

C++

```
BS_BOOL bUnderrunProtection;
```

**Description**

You can activate the buffer-underrun-protection by setting the value to BS\_TRUE (see page 499). BS\_FALSE (see page 335) de-activates the buffer protection.

### 1.1.24.3.11 SOptions::bVerifyAfterBurn Data Member

C++

```
BS_BOOL bVerifyAfterBurn;
```

**Description**

This argument enables or disables the verify after burning. This option is only available for Data CD/DVD/BD ISO and UDF projects.

### 1.1.24.3.12 SOptions::chBootImage Data Member

C++

```
TCHAR chBootImage[_MAX_PATH];
```

**Description**

File name of the image (with path information)

### 1.1.24.3.13 SOptions::chVolumeLabel Data Member

C++

```
TCHAR chVolumeLabel[128];
```

**Description**

This argument sets the volume name of the medium (null terminated). This name will be displayed e.g. in the Windows Explorer. This string may neither contain high characters nor spaces.

### 1.1.24.3.14 SOptions::nCacheSize Data Member

C++

```
int32 nCacheSize;
```

**Description**

Sets the size of the burning buffer to be used (in bytes; default: 4 \* 1024 \* 1024 [4 MB])

We recommend to use a range between 4 MB and 32 MB.

4 MB = CD

16 MB = DVD

32 MB = BD

### 1.1.24.3.15 SOptions::nCopies Data Member

C++

```
int32 nCopies;
```

**Description**

This argument sets the number of copies to burn. Only available in GUI mode.

### 1.1.24.3.16 SOptions::nWriteMethod Data Member

C++

```
int32 nWriteMethod;
```

**Description**

This parameter sets the burning method. This option is only allowed for CD-R/CD-RW media. For DVD media it is ignored.  
Values are:

BS\_WM\_TAO (see page 507),  
BS\_WM\_DAO (see page 506),  
BS\_WM.DAO96 (see page 506)

## 1.1.24.4 SOptions Methods

The methods of the SOptions class are listed here.

**Public Methods**

	Name	Description
Empty	Empty (see page 82)	This is Empty, a member of class SOptions.
INSERT_STRUCTURE_PADDING	(see page 82)	This is INSERT_STRUCTURE_PADDING, a member of class SOptions.

### 1.1.24.4.1 SOptions::Empty Method

**C++**

```
void Empty();
```

**Description**

This is Empty, a member of class SOptions.

### 1.1.24.4.2 SOptions::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA - sizeof(BS_BOOL));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SOptions.

## 1.1.25 SRAWTrackToAdd Structure

**C++**

```
struct SRAWTrackToAdd {
    int32 nNumber;
    int32 nIndex;
    int32 nFormat;
    int32 nDataTypeMask;
    int32 nIgnoreDataMask;
    int32 nStartAddress;
    int32 nLength;
    int64 nOffset;
};
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK sub class that contains the information of a RAW image.

## 1.1.25.1 SRAWTrackToAdd Members

The following tables list the members exposed by SRAWTrackToAdd.

**Public Data Members**

	Name	Description
◆	nDataTypeMask ( <a href="#">see page 84</a> )	The format of the data. This can be a combination of the following values: BS_RDT_SYNC_HEADER ( <a href="#">see page 370</a> ), BS_RDT_SUBHEADERS ( <a href="#">see page 370</a> ), BS_RDT_DATA ( <a href="#">see page 369</a> ), BS_RDT_EDC_ECC ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_PQ ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_PW ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_RW ( <a href="#">see page 370</a> )
◆	nFormat ( <a href="#">see page 84</a> )	The format of the track. This can be one of the following values: BS_RTF_AUDIO ( <a href="#">see page 377</a> ), BS_RTF_MODE1 ( <a href="#">see page 377</a> ), BS_RTF_MODE2_FORMLESS ( <a href="#">see page 378</a> ), BS_RTF_MODE2_FORM1 ( <a href="#">see page 377</a> ), BS_RTF_MODE2_FORM2 ( <a href="#">see page 378</a> )
◆	nIgnoreDataMask ( <a href="#">see page 85</a> )	Format of the data to be ignored. See nDataTypeMask ( <a href="#">see page 84</a> ) description.
◆	nIndex ( <a href="#">see page 85</a> )	Index inside the track. 0 - pre-gap, 1..99 - user data
◆	nLength ( <a href="#">see page 85</a> )	The length of the track in sectors.
◆	nNumber ( <a href="#">see page 85</a> )	The number of the track. 0 - Lead-in, 0xAA – lead-out. 1 – first track, 2 – second track ...
◆	nOffset ( <a href="#">see page 85</a> )	The offset of the data in the image file in bytes. Not used if burning through callback function.
◆	nStartAddress ( <a href="#">see page 85</a> )	This is the start address of the track on the disc in sectors.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 86</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SRAWTrackToAdd.
◆	SRAWTrackToAdd ( <a href="#">see page 83</a> )	This is SRAWTrackToAdd, a member of class SRAWTrackToAdd.

## 1.1.25.2 SRAWTrackToAdd::SRAWTrackToAdd Constructor

**C++**

```
SRAWTrackToAdd();
```

**Description**

This is SRAWTrackToAdd, a member of class SRAWTrackToAdd.

## 1.1.25.3 SRAWTrackToAdd Data Members

The data members of the SRAWTrackToAdd class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	nDataTypeMask ( <a href="#">see page 84</a> )	The format of the data. This can be a combination of the following values: BS_RDT_SYNC_HEADER ( <a href="#">see page 370</a> ), BS_RDT_SUBHEADERS ( <a href="#">see page 370</a> ), BS_RDT_DATA ( <a href="#">see page 369</a> ), BS_RDT_EDC_ECC ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_PQ ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_PW ( <a href="#">see page 369</a> ), BS_RDT_SUBCH_RW ( <a href="#">see page 370</a> )
◆	nFormat ( <a href="#">see page 84</a> )	The format of the track. This can be one of the following values: BS_RTF_AUDIO ( <a href="#">see page 377</a> ), BS_RTF_MODE1 ( <a href="#">see page 377</a> ), BS_RTF_MODE2_FORMLESS ( <a href="#">see page 378</a> ), BS_RTF_MODE2_FORM1 ( <a href="#">see page 377</a> ), BS_RTF_MODE2_FORM2 ( <a href="#">see page 378</a> )
◆	nIgnoreDataMask ( <a href="#">see page 85</a> )	Format of the data to be ignored. See nDataTypeMask ( <a href="#">see page 84</a> ) description.
◆	nIndex ( <a href="#">see page 85</a> )	Index inside the track. 0 - pre-gap, 1..99 - user data
◆	nLength ( <a href="#">see page 85</a> )	The length of the track in sectors.
◆	nNumber ( <a href="#">see page 85</a> )	The number of the track. 0 - Lead-in, 0xAA – lead-out. 1 – first track, 2 – second track ...
◆	nOffset ( <a href="#">see page 85</a> )	The offset of the data in the image file in bytes. Not used if burning through callback function.
◆	nStartAddress ( <a href="#">see page 85</a> )	This is the start address of the track on the disc in sectors.

**1.1.25.3.1 SRAWTrackToAdd::nDataTypeMask Data Member****C++**

```
int32 nDataTypeMask;
```

**Description**

The format of the data. This can be a combination of the following values:

BS\_RDT\_SYNC\_HEADER ([see page 370](#)),  
BS\_RDT\_SUBHEADERS ([see page 370](#)),  
BS\_RDT\_DATA ([see page 369](#)),  
BS\_RDT\_EDC\_ECC ([see page 369](#)),  
BS\_RDT\_SUBCH\_PQ ([see page 369](#)),  
BS\_RDT\_SUBCH\_PW ([see page 369](#)),  
BS\_RDT\_SUBCH\_RW ([see page 370](#))

**1.1.25.3.2 SRAWTrackToAdd::nFormat Data Member****C++**

```
int32 nFormat;
```

**Description**

The format of the track. This can be one of the following values:

BS\_RTF\_AUDIO ([see page 377](#)),

BS\_RTF\_MODE1 (see page 377),  
BS\_RTF\_MODE2\_FORMLESS (see page 378),  
BS\_RTF\_MODE2\_FORM1 (see page 377),  
BS\_RTF\_MODE2\_FORM2 (see page 378)

### 1.1.25.3.3 SRAWTrackToAdd::nIgnoreDataMask Data Member

#### C++

```
int32 nIgnoreDataMask;
```

#### Description

Format of the data to be ignored. See nDataTypeMask (see page 84) description.

### 1.1.25.3.4 SRAWTrackToAdd::nIndex Data Member

#### C++

```
int32 nIndex;
```

#### Description

Index inside the track. 0 - pre-gap, 1..99 - user data

### 1.1.25.3.5 SRAWTrackToAdd::nLength Data Member

#### C++

```
int32 nLength;
```

#### Description

The length of the track in sectors.

### 1.1.25.3.6 SRAWTrackToAdd::nNumber Data Member

#### C++

```
int32 nNumber;
```

#### Description

The number of the track. 0 - Lead-in, 0xAA – lead-out. 1 – first track, 2 – second track ...

### 1.1.25.3.7 SRAWTrackToAdd::nOffset Data Member

#### C++

```
int64 nOffset;
```

#### Description

The offset of the data in the image file in bytes. Not used if burning through callback function.

### 1.1.25.3.8 SRAWTrackToAdd::nStartAddress Data Member

#### C++

```
int32 nStartAddress;
```

#### Description

This is the start address of the track on the disc in sectors.

## 1.1.25.4 SRAWTrackToAdd Methods

The methods of the SRAWTrackToAdd class are listed here.

### Public Methods

	Name	Description
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 86</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SRAWTrackToAdd.

### 1.1.25.4.1 SRAWTrackToAdd::INSERT\_STRUCTURE\_PADDING Method

#### C++

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL);
```

#### Description

This is INSERT\_STRUCTURE\_PADDING, a member of class SRAWTrackToAdd.

## 1.1.26 SReadErrorCorrectionParams Structure

#### C++

```
struct SReadErrorCorrectionParams {
    BS_BOOL bErrorCorrection;
    BS_BOOL bBlankBadSectors;
    uint8 nHardwareRetryCount;
    uint8 nSoftwareRetryCount;
};
```

#### File

IsoSDKDefinitions.h ([see page 517](#))

#### Description

IsoSDK sub class that contains the information of the ReadErrorCorrection for DiskCopy and CreateImage ([see page 127](#)).

## 1.1.26.1 SReadErrorCorrectionParams Members

The following tables list the members exposed by SReadErrorCorrectionParams.

### Public Data Members

	Name	Description
❖	bBlankBadSectors ( <a href="#">see page 87</a> )	A BOOL value that indicate to fill bad sectors with blank sectors on the target disc or image. This will help to keep the original size and positions.
❖	bErrorCorrection ( <a href="#">see page 87</a> )	A BOOL value that indicates to use the software and hardware retry functions.
❖	nHardwareRetryCount ( <a href="#">see page 87</a> )	This value indicates how many times the device try to read possible bad sectors.
❖	nSoftwareRetryCount ( <a href="#">see page 88</a> )	This value indicates how many times the software try to read possible bad sectors.

### Public Methods

	Name	Description
❖	INSERT_STRUCTURE_PADDING ( <a href="#">see page 88</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SReadErrorCorrectionParams.

	SReadErrorCorrectionParams ( <a href="#">see page 87</a> )	This is SReadErrorCorrectionParams, a member of class SReadErrorCorrectionParams.
---	--	---

## 1.1.26.2 SReadErrorCorrectionParams::SReadErrorCorrectionParams Constructor

C++

```
SReadErrorCorrectionParams();
```

### Description

This is SReadErrorCorrectionParams, a member of class SReadErrorCorrectionParams.

## 1.1.26.3 SReadErrorCorrectionParams Data Members

The data members of the SReadErrorCorrectionParams class are listed here.

### Public Data Members

	Name	Description
◆	bBlankBadSectors ( <a href="#">see page 87</a> )	A BOOL value that indicate to fill bad sectors with blank sectors on the target disc or image. This will help to keep the original size and positions.
◆	bErrorCorrection ( <a href="#">see page 87</a> )	A BOOL value that indicates to use the software and hardware retry functions.
◆	nHardwareRetryCount ( <a href="#">see page 87</a> )	This value indicates how many times the device try to read possible bad sectors.
◆	nSoftwareRetryCount ( <a href="#">see page 88</a> )	This value indicates how many times the software try to read possible bad sectors.

### 1.1.26.3.1 SReadErrorCorrectionParams::bBlankBadSectors Data Member

C++

```
BS_BOOL bBlankBadSectors;
```

### Description

A BOOL value that indicate to fill bad sectors with blank sectors on the target disc or image. This will help to keep the original size and positions.

### 1.1.26.3.2 SReadErrorCorrectionParams::bErrorCorrection Data Member

C++

```
BS_BOOL bErrorCorrection;
```

### Description

A BOOL value that indicates to use the software and hardware retry functions.

### 1.1.26.3.3 SReadErrorCorrectionParams::nHardwareRetryCount Data Member

C++

```
uint8 nHardwareRetryCount;
```

**Description**

This value indicates how many times the device try to read possible bad sectors.

### 1.1.26.3.4 SReadErrorCorrectionParams::nSoftwareRetryCount Data Member

**C++**

```
uint8 nSoftwareRetryCount;
```

**Description**

This value indicates how many times the software try to read possible bad sectors.

### 1.1.26.4 SReadErrorCorrectionParams Methods

The methods of the SReadErrorCorrectionParams class are listed here.

**Public Methods**

	Name	Description
!	INSERT_STRUCTURE_PADDING ( <a href="#">see page 88</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SReadErrorCorrectionParams.

#### 1.1.26.4.1 SReadErrorCorrectionParams::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SReadErrorCorrectionParams.

---

### 1.1.27 SSessionInfo Structure

**C++**

```
struct SSessionInfo {
    long dSessionSize;
    BS_BOOL bLastSession;
    long dStartLBA;
    long nFirstTrack;
    long nLastTrack;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the selected session.

#### 1.1.27.1 SSessionInfo Members

The following tables list the members exposed by SSessionInfo.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	bLastSession ( <a href="#">see page 89</a> )	Is it the last disc session.
◆	dSessionSize ( <a href="#">see page 89</a> )	Session size in sectors.
◆	dStartLBA ( <a href="#">see page 90</a> )	Address of the first session's sector.
◆	nFirstTrack ( <a href="#">see page 90</a> )	Number of the first track of the session.
◆	nLastTrack ( <a href="#">see page 90</a> )	Number of the last track of the session.

**Public Methods**

	<b>Name</b>	<b>Description</b>
≡	INSERT_STRUCTURE_PADDING ( <a href="#">see page 90</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SSessionInfo.
≡	SSessionInfo ( <a href="#">see page 89</a> )	This is SSessionInfo, a member of class SSessionInfo.

## 1.1.27.2 SSessionInfo::SSessionInfo Constructor

**C++**

```
SSessionInfo();
```

**Description**

This is SSessionInfo, a member of class SSessionInfo.

## 1.1.27.3 SSessionInfo Data Members

The data members of the SSessionInfo class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	bLastSession ( <a href="#">see page 89</a> )	Is it the last disc session.
◆	dSessionSize ( <a href="#">see page 89</a> )	Session size in sectors.
◆	dStartLBA ( <a href="#">see page 90</a> )	Address of the first session's sector.
◆	nFirstTrack ( <a href="#">see page 90</a> )	Number of the first track of the session.
◆	nLastTrack ( <a href="#">see page 90</a> )	Number of the last track of the session.

### 1.1.27.3.1 SSessionInfo::bLastSession Data Member

**C++**

```
BS_BOOL bLastSession;
```

**Description**

Is it the last disc session.

### 1.1.27.3.2 SSessionInfo::dSessionSize Data Member

**C++**

```
long dSessionSize;
```

**Description**

Session size in sectors.

### 1.1.27.3.3 SSessionInfo::dStartLBA Data Member

**C++**

```
long dStartLBA;
```

**Description**

Address of the first session's sector.

### 1.1.27.3.4 SSessionInfo::nFirstTrack Data Member

**C++**

```
long nFirstTrack;
```

**Description**

Number of the first track of the session.

### 1.1.27.3.5 SSessionInfo::nLastTrack Data Member

**C++**

```
long nLastTrack;
```

**Description**

Number of the last track of the session.

## 1.1.27.4 SSessionInfo Methods

The methods of the SSessionInfo class are listed here.

**Public Methods**

	Name	Description
☞	INSERT_STRUCTURE_PADDING (☞ see page 90)	This is INSERT_STRUCTURE_PADDING, a member of class SSessionInfo.

### 1.1.27.4.1 SSessionInfo::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_BIG);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SSessionInfo.

---

## 1.1.28 SSspeed Structure

**C++**

```
struct SSspeed {
    float fSpeed;
    int nSpeedInKBPerSec;
};
```

**File**

IsoSDKDefinitions.h (☞ see page 517)

**Description**

IsoSDK sub class that contains information about the possible burning speeds.

## 1.1.28.1 SSspeed Members

The following tables list the members exposed by SSspeed.

**Public Data Members**

	Name	Description
◆	fSpeed ( <a href="#">see page 91</a> )	Values of normal speed information, like 32.0 or 48.0
◆	nSpeedInKBPerSec ( <a href="#">see page 91</a> )	The original internal speed, like 4800 (32x) or 7200 (48x)

**Public Methods**

	Name	Description
≡	INSERT_STRUCTURE_PADDING ( <a href="#">see page 92</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SSspeed.
≡	SSpeed ( <a href="#">see page 91</a> )	This is SSspeed, a member of class SSspeed.

## 1.1.28.2 SSspeed::SSpeed Constructor

**C++**

```
SSpeed();
```

**Description**

This is SSspeed, a member of class SSspeed.

## 1.1.28.3 SSspeed Data Members

The data members of the SSspeed class are listed here.

**Public Data Members**

	Name	Description
◆	fSpeed ( <a href="#">see page 91</a> )	Values of normal speed information, like 32.0 or 48.0
◆	nSpeedInKBPerSec ( <a href="#">see page 91</a> )	The original internal speed, like 4800 (32x) or 7200 (48x)

### 1.1.28.3.1 SSspeed::fSpeed Data Member

**C++**

```
float fSpeed;
```

**Description**

Values of normal speed information, like 32.0 or 48.0

### 1.1.28.3.2 SSspeed::nSpeedInKBPerSec Data Member

**C++**

```
int nSpeedInKBPerSec;
```

**Description**

The original internal speed, like 4800 (32x) or 7200 (48x)

## 1.1.28.4 SSspeed Methods

The methods of the SSspeed class are listed here.

**Public Methods**

	Name	Description
!	INSERT_STRUCTURE_PADDING ( <a href="#">see page 92</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SSspeed.

### 1.1.28.4.1 SSspeed::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_SMALL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SSspeed.

## 1.1.29 STags Structure

**C++**

```
struct STags {
    TCHAR chArtist[128];
    TCHAR chAlbum[128];
    TCHAR chTitle[128];
    TCHAR chComment[255];
};
```

**File**

IsoSDKDefinitions.h ( [see page 517](#) )

**Description**

IsoSDK sub class that

### 1.1.29.1 STags Members

The following tables list the members exposed by STags.

**Public Data Members**

	Name	Description
!	chAlbum ( <a href="#">see page 93</a> )	This is chAlbum, a member of class STags.
!	chArtist ( <a href="#">see page 93</a> )	This is chArtist, a member of class STags.
!	chComment ( <a href="#">see page 93</a> )	This is chComment, a member of class STags.
!	chTitle ( <a href="#">see page 94</a> )	This is chTitle, a member of class STags.

**Public Methods**

	Name	Description
Empty ( <a href="#">see page 94</a> )		This is Empty, a member of class STags.
STags ( <a href="#">see page 93</a> )		IsoSDK sub class that

## 1.1.29.2 STags::STags Constructor

**C++**

```
STags();
```

**Description**

IsoSDK sub class that

## 1.1.29.3 STags Data Members

The data members of the STags class are listed here.

**Public Data Members**

	Name	Description
◆	chAlbum ( <a href="#">see page 93</a> )	This is chAlbum, a member of class STags.
◆	chArtist ( <a href="#">see page 93</a> )	This is chArtist, a member of class STags.
◆	chComment ( <a href="#">see page 93</a> )	This is chComment, a member of class STags.
◆	chTitle ( <a href="#">see page 94</a> )	This is chTitle, a member of class STags.

### 1.1.29.3.1 STags::chAlbum Data Member

**C++**

```
TCHAR chAlbum[128];
```

**Description**

This is chAlbum, a member of class STags.

### 1.1.29.3.2 STags::chArtist Data Member

**C++**

```
TCHAR chArtist[128];
```

**Description**

This is chArtist, a member of class STags.

### 1.1.29.3.3 STags::chComment Data Member

**C++**

```
TCHAR chComment[255];
```

**Description**

This is chComment, a member of class STags.

### 1.1.29.3.4 STags::chTitle Data Member

**C++**

```
TCHAR chTitle[128];
```

**Description**

This is chTitle, a member of class STags.

### 1.1.29.4 STags Methods

The methods of the STags class are listed here.

**Public Methods**

	Name	Description
Empty	Empty ( <a href="#">see page 94</a> )	This is Empty, a member of class STags.

#### 1.1.29.4.1 STags::Empty Method

**C++**

```
void Empty();
```

**Description**

This is Empty, a member of class STags.

---

### 1.1.30 STrackInfo Structure

**C++**

```
struct STrackInfo {
    int16 nTrackNumber;
    int16 nSessionNumber;
    int32 nStartLBA;
    int32 nSize;
    int8 nFormat;
    int8 nFileSystem;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the selected track.

#### 1.1.30.1 STrackInfo Members

The following tables list the members exposed by STrackInfo.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	nFileSystem ( <a href="#">see page 96</a> )	The file system of the track. A combination of the following flags will be returned: BS_FS_UNKNOWN ( <a href="#">see page 338</a> ), BS_FS_ISO9660 ( <a href="#">see page 337</a> ), BS_FS_JOLIET ( <a href="#">see page 337</a> ), BS_FS_UDF ( <a href="#">see page 337</a> ), BS_FS_BOOTABLE ( <a href="#">see page 336</a> )
◆	nFormat ( <a href="#">see page 96</a> )	The format of the track. The following values will be returned: BS_TF_AUDIO ( <a href="#">see page 497</a> ), BS_TF_DATA_MODE1 ( <a href="#">see page 497</a> ), BS_TF_DATA_MODE2 ( <a href="#">see page 497</a> )
◆	nSessionNumber ( <a href="#">see page 96</a> )	Session number of track.
◆	nSize ( <a href="#">see page 96</a> )	The track size in bytes.
◆	nStartLBA ( <a href="#">see page 96</a> )	Track's start address.
◆	nTrackNumber ( <a href="#">see page 97</a> )	The track number. you want to receive information from.

**Public Methods**

	<b>Name</b>	<b>Description</b>
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 97</a> )	This is INSERT_STRUCTURE_PADDING, a member of class STrackInfo.
◆	STrackInfo ( <a href="#">see page 95</a> )	This is STrackInfo, a member of class STrackInfo.

## 1.1.30.2 STrackInfo::STrackInfo Constructor

**C++**

```
STrackInfo();
```

**Description**

This is STrackInfo, a member of class STrackInfo.

## 1.1.30.3 STrackInfo Data Members

The data members of the STrackInfo class are listed here.

**Public Data Members**

	<b>Name</b>	<b>Description</b>
◆	nFileSystem ( <a href="#">see page 96</a> )	The file system of the track. A combination of the following flags will be returned: BS_FS_UNKNOWN ( <a href="#">see page 338</a> ), BS_FS_ISO9660 ( <a href="#">see page 337</a> ), BS_FS_JOLIET ( <a href="#">see page 337</a> ), BS_FS_UDF ( <a href="#">see page 337</a> ), BS_FS_BOOTABLE ( <a href="#">see page 336</a> )
◆	nFormat ( <a href="#">see page 96</a> )	The format of the track. The following values will be returned: BS_TF_AUDIO ( <a href="#">see page 497</a> ), BS_TF_DATA_MODE1 ( <a href="#">see page 497</a> ), BS_TF_DATA_MODE2 ( <a href="#">see page 497</a> )
◆	nSessionNumber ( <a href="#">see page 96</a> )	Session number of track.
◆	nSize ( <a href="#">see page 96</a> )	The track size in bytes.
◆	nStartLBA ( <a href="#">see page 96</a> )	Track's start address.



nTrackNumber (see page 97)

The track number. you want to receive information from.

### 1.1.30.3.1 STrackInfo::nFileSystem Data Member

**C++**

```
int8 nFileSystem;
```

**Description**

The file system of the track. A combination of the following flags will be returned:

BS\_FS\_UNKNOWN (see page 338),  
BS\_FS\_ISO9660 (see page 337),  
BS\_FS\_JOLIET (see page 337),  
BS\_FS\_UDF (see page 337),  
BS\_FS\_BOOTABLE (see page 336)

### 1.1.30.3.2 STrackInfo::nFormat Data Member

**C++**

```
int8 nFormat;
```

**Description**

The format of the track. The following values will be returned:

BS\_TF\_AUDIO (see page 497),  
BS\_TF\_DATA\_MODE1 (see page 497),  
BS\_TF\_DATA\_MODE2 (see page 497)

### 1.1.30.3.3 STrackInfo::nSessionNumber Data Member

**C++**

```
int16 nSessionNumber;
```

**Description**

Session number of track.

### 1.1.30.3.4 STrackInfo::nSize Data Member

**C++**

```
int32 nSize;
```

**Description**

The track size in bytes.

### 1.1.30.3.5 STrackInfo::nStartLBA Data Member

**C++**

```
int32 nStartLBA;
```

**Description**

Track's start address.

### 1.1.30.3.6 STrackInfo::nTrackNumber Data Member

**C++**

```
int16 nTrackNumber;
```

**Description**

The track number. you want to receive information from.

### 1.1.30.4 STrackInfo Methods

The methods of the STrackInfo class are listed here.

**Public Methods**

	Name	Description
!	INSERT_STRUCTURE_PADDING ( <a href="#">see page 97</a> )	This is INSERT_STRUCTURE_PADDING, a member of class STrackInfo.

#### 1.1.30.4.1 STrackInfo::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class STrackInfo.

---

### 1.1.31 SUDFOptions Structure

**C++**

```
struct SUDFOptions {
    int32 Version;
    int32 PartitionType;
    BS_BOOL WriteFileStreams;
    BS_BOOL AvchdDisc;
};
```

**File**

IsoSDKDefinitions.h ( [see page 517](#) )

**Description**

IsoSDK sub class that is used to get and set the parameters for UDF disc.

#### 1.1.31.1 SUDFOptions Members

The following tables list the members exposed by SUDFOptions.

**Public Data Members**

	Name	Description
!	AvchdDisc ( <a href="#">see page 98</a> )	Allow to burn the Blu-ray file system to a DVD blank.

◆	PartitionType ( <a href="#">see page 99</a> )	UDF partition type. Possible Values are: BS_UDF_PARTITION_PHYSICAL ( <a href="#">see page 499</a> ), BS_UDF_PARTITION_VIRTUAL ( <a href="#">see page 500</a> ), BS_UDF_PARTITION_SPARABLE ( <a href="#">see page 499</a> )
◆	Version ( <a href="#">see page 99</a> )	UDF version. Possible values are: BS_UDF_VERSION_102 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_150 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_200 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_201 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_250 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_260 ( <a href="#">see page 501</a> )
◆	WriteFileStreams ( <a href="#">see page 99</a> )	States whether file streams should be used.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 100</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SUDFOptions.
◆	SUDFOptions ( <a href="#">see page 98</a> )	This is SUDFOptions, a member of class SUDFOptions.

## 1.1.31.2 SUDFOptions::SUDFOptions Constructor

**C++**

```
SUDFOptions();
```

**Description**

This is SUDFOptions, a member of class SUDFOptions.

## 1.1.31.3 SUDFOptions Data Members

The data members of the SUDFOptions class are listed here.

**Public Data Members**

	Name	Description
◆	AvchdDisc ( <a href="#">see page 98</a> )	Allow to burn the Blu-ray file system to a DVD blank.
◆	PartitionType ( <a href="#">see page 99</a> )	UDF partition type. Possible Values are: BS_UDF_PARTITION_PHYSICAL ( <a href="#">see page 499</a> ), BS_UDF_PARTITION_VIRTUAL ( <a href="#">see page 500</a> ), BS_UDF_PARTITION_SPARABLE ( <a href="#">see page 499</a> )
◆	Version ( <a href="#">see page 99</a> )	UDF version. Possible values are: BS_UDF_VERSION_102 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_150 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_200 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_201 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_250 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_260 ( <a href="#">see page 501</a> )
◆	WriteFileStreams ( <a href="#">see page 99</a> )	States whether file streams should be used.

### 1.1.31.3.1 SUDFOptions::AvchdDisc Data Member

**C++**

```
BS_BOOL AvchdDisc;
```

**Description**

Allow to burn the Blu-ray file system to a DVD blank.

**1.1.31.3.2 SUDFOptions::PartitionType Data Member****C++**

```
int32 PartitionType;
```

**Description**

UDF partition type. Possible Values are:

BS\_UDF\_PARTITION\_PHYSICAL ([see page 499](#)),  
 BS\_UDF\_PARTITION\_VIRTUAL ([see page 500](#)),  
 BS\_UDF\_PARTITION\_SPARABLE ([see page 499](#))

**1.1.31.3.3 SUDFOptions::Version Data Member****C++**

```
int32 Version;
```

**Description**

UDF version. Possible values are:

BS\_UDF\_VERSION\_102 ([see page 500](#)),  
 BS\_UDF\_VERSION\_150 ([see page 500](#)),  
 BS\_UDF\_VERSION\_200 ([see page 501](#)),  
 BS\_UDF\_VERSION\_201 ([see page 501](#)),  
 BS\_UDF\_VERSION\_250 ([see page 501](#)),  
 BS\_UDF\_VERSION\_260 ([see page 501](#))

**1.1.31.3.4 SUDFOptions::WriteFileStreams Data Member****C++**

```
BS_BOOL WriteFileStreams;
```

**Description**

States whether file streams should be used.

**1.1.31.4 SUDFOptions Methods**

The methods of the SUDFOptions class are listed here.

**Public Methods**

	Name	Description
	INSERT_STRUCTURE_PADDING ( <a href="#">see page 100</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SUDFOptions.

### 1.1.31.4.1 SUDFOptions::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL - sizeof(BS_BOOL));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SUDFOptions.

## 1.1.32 SUDFVolumeInfo Structure

**C++**

```
struct SUDFVolumeInfo {
    TCHAR chVolumeLabel[128];
    TCHAR chPreparer[128];
    int32 nVersion;
    int32 nPartitionType;
    int32 nMVDSAddress;
    int32 nRVDSAddress;
    int32 nRootAddress;
    int32 nRootFEAddress;
    int32 nPartitionAddress;
    int32 nPartitionLength;
    int32 nLVDAddress;
    int32 nPVDAddress;
    int32 nFSDAddress;
    int32 nVATAAddress;
    int32 nMetadataAddress;
    int32 nSparingAddress;
    int32 nFileCount;
    int32 nDirCount;
    SFileDateTime tRecordingDateTime;
};
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains the information about the UDF information.

### 1.1.32.1 SUDFVolumeInfo Members

The following tables list the members exposed by SUDFVolumeInfo.

**Public Data Members**

	Name	Description
❖	chPreparer ( <a href="#">see page 102</a> )	The name of the data preparer.
❖	chVolumeLabel ( <a href="#">see page 102</a> )	The name of the UDF volume label.
❖	nDirCount ( <a href="#">see page 102</a> )	The number of directories inside the UDF file system.
❖	nFileCount ( <a href="#">see page 102</a> )	The number of files inside the UDF file system.
❖	nFSDAddress ( <a href="#">see page 103</a> )	File system descriptor address.
❖	nLVDAddress ( <a href="#">see page 103</a> )	Logical volume descriptor address.
❖	nMetadataAddress ( <a href="#">see page 103</a> )	Metadata partition address.
❖	nMVDSAddress ( <a href="#">see page 103</a> )	Main volume descriptor sequence address.
❖	nPartitionAddress ( <a href="#">see page 103</a> )	Partition address.

◆	nPartitionLength (see page 103)	Partition length.
◆	nPartitionType (see page 103)	The partition type of the UDF filesystem. Possible values are: BS_UDF_PARTITION_PHYSICAL (see page 499), BS_UDF_PARTITION_VIRTUAL (see page 500), BS_UDF_PARTITION_SPARABLE (see page 499)
◆	nPVDAddress (see page 104)	Partition volume descriptor address.
◆	nRootAddress (see page 104)	Root directory extent address.
◆	nRootFEAddress (see page 104)	Root directory file entry address.
◆	nRVDSAddress (see page 104)	Reserved volume descriptor sequence address.
◆	nSparingAddress (see page 104)	Sparing area address.
◆	nVATAddress (see page 104)	Virtual allocation table address.
◆	nVersion (see page 105)	The version of the UDF filesystem. Possible values are: BS_UDF_VERSION_102 (see page 500), BS_UDF_VERSION_150 (see page 500), BS_UDF_VERSION_200 (see page 501), BS_UDF_VERSION_201 (see page 501), BS_UDF_VERSION_250 (see page 501), BS_UDF_VERSION_260 (see page 501)
◆ <small>New</small>	tRecordingDateTime (see page 105)	The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

**Public Methods**

	Name	Description
◆	INSERT_STRUCTURE_PADDING (see page 105)	This is INSERT_STRUCTURE_PADDING, a member of class SUDFVolumeInfo.
◆	SUDFVolumeInfo (see page 101)	This is SUDFVolumeInfo, a member of class SUDFVolumeInfo.

## 1.1.32.2 SUDFVolumeInfo::SUDFVolumeInfo Constructor

**C++**

```
SUDFVolumeInfo();
```

**Description**

This is SUDFVolumeInfo, a member of class SUDFVolumeInfo.

## 1.1.32.3 SUDFVolumeInfo Data Members

The data members of the SUDFVolumeInfo class are listed here.

**Public Data Members**

	Name	Description
◆	chPreparer (see page 102)	The name of the data preparer.
◆	chVolumeLabel (see page 102)	The name of the UDF volume label.
◆	nDirCount (see page 102)	The number of directories inside the UDF file system.
◆	nFileCount (see page 102)	The number of files inside the UDF file system.
◆	nFSDAddress (see page 103)	File system descriptor address.
◆	nLVDAAddress (see page 103)	Logical volume descriptor address.
◆	nMetadataAddress (see page 103)	Metadata partition address.
◆	nMVDSAddress (see page 103)	Main volume descriptor sequence address.
◆	nPartitionAddress (see page 103)	Partition address.

◆	nPartitionLength ( <a href="#">see page 103</a> )	Partition length.
◆	nPartitionType ( <a href="#">see page 103</a> )	The partition type of the UDF filesystem. Possible values are: BS_UDF_PARTITION_PHYSICAL ( <a href="#">see page 499</a> ), BS_UDF_PARTITION_VIRTUAL ( <a href="#">see page 500</a> ), BS_UDF_PARTITION_SPARABLE ( <a href="#">see page 499</a> )
◆	nPVDAddress ( <a href="#">see page 104</a> )	Partition volume descriptor address.
◆	nRootAddress ( <a href="#">see page 104</a> )	Root directory extent address.
◆	nRootFEAddress ( <a href="#">see page 104</a> )	Root directory file entry address.
◆	nRVDSAddress ( <a href="#">see page 104</a> )	Reserved volume descriptor sequence address.
◆	nSparingAddress ( <a href="#">see page 104</a> )	Sparing area address.
◆	nVATAddress ( <a href="#">see page 104</a> )	Virtual allocation table address.
◆	nVersion ( <a href="#">see page 105</a> )	The version of the UDF filesystem. Possible values are: BS_UDF_VERSION_102 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_150 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_200 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_201 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_250 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_260 ( <a href="#">see page 501</a> )
◆ <small>new</small>	tRecordingDateTime ( <a href="#">see page 105</a> )	The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

### 1.1.32.3.1 SUDFVolumeInfo::chPreparer Data Member

C++

```
TCHAR chPreparer[128];
```

**Description**

The name of the data preparer.

### 1.1.32.3.2 SUDFVolumeInfo::chVolumeLabel Data Member

C++

```
TCHAR chVolumeLabel[128];
```

**Description**

The name of the UDF volume label.

### 1.1.32.3.3 SUDFVolumeInfo::nDirCount Data Member

C++

```
int32 nDirCount;
```

**Description**

The number of directories inside the UDF file system.

### 1.1.32.3.4 SUDFVolumeInfo::nFileCount Data Member

C++

```
int32 nFileCount;
```

**Description**

The number of files inside the UDF file system.

### 1.1.32.3.5 SUDFVolumeInfo::nFSDAddress Data Member

C++

```
int32 nFSDAddress;
```

Description

File system descriptor address.

### 1.1.32.3.6 SUDFVolumeInfo::nLVDAAddress Data Member

C++

```
int32 nLVDAAddress;
```

Description

Logical volume descriptor address.

### 1.1.32.3.7 SUDFVolumeInfo::nMetadataAddress Data Member

C++

```
int32 nMetadataAddress;
```

Description

Metadata partition address.

### 1.1.32.3.8 SUDFVolumeInfo::nMVDSAddress Data Member

C++

```
int32 nMVDSAddress;
```

Description

Main volume descriptor sequence address.

### 1.1.32.3.9 SUDFVolumeInfo::nPartitionAddress Data Member

C++

```
int32 nPartitionAddress;
```

Description

Partition address.

### 1.1.32.3.10 SUDFVolumeInfo::nPartitionLength Data Member

C++

```
int32 nPartitionLength;
```

Description

Partition length.

### 1.1.32.3.11 SUDFVolumeInfo::nPartitionType Data Member

C++

```
int32 nPartitionType;
```

**Description**

The partition type of the UDF filesystem. Possible values are:

BS\_UDF\_PARTITION\_PHYSICAL (see page 499),

BS\_UDF\_PARTITION\_VIRTUAL (see page 500),

BS\_UDF\_PARTITION\_SPARABLE (see page 499)

### 1.1.32.3.12 SUDFVolumeInfo::nPVDAddress Data Member

**C++**

```
int32 nPVDAddress;
```

**Description**

Partition volume descriptor address.

### 1.1.32.3.13 SUDFVolumeInfo::nRootAddress Data Member

**C++**

```
int32 nRootAddress;
```

**Description**

Root directory extent address.

### 1.1.32.3.14 SUDFVolumeInfo::nRootFEAddress Data Member

**C++**

```
int32 nRootFEAddress;
```

**Description**

Root directory file entry address.

### 1.1.32.3.15 SUDFVolumeInfo::nRVDSAddress Data Member

**C++**

```
int32 nRVDSAddress;
```

**Description**

Reserved volume descriptor sequence address.

### 1.1.32.3.16 SUDFVolumeInfo::nSparingAddress Data Member

**C++**

```
int32 nSparingAddress;
```

**Description**

Sparing area address.

### 1.1.32.3.17 SUDFVolumeInfo::nVATAddress Data Member

**C++**

```
int32 nVATAddress;
```

**Description**

Virtual allocation table address.

**1.1.32.3.18 SUDFVolumeInfo::nVersion Data Member****C++**

```
int32 nVersion;
```

**Description**

The version of the UDF filesystem. Possible values are:

BS\_UDF\_VERSION\_102 (see page 500),  
 BS\_UDF\_VERSION\_150 (see page 500),  
 BS\_UDF\_VERSION\_200 (see page 501),  
 BS\_UDF\_VERSION\_201 (see page 501),  
 BS\_UDF\_VERSION\_250 (see page 501),  
 BS\_UDF\_VERSION\_260 (see page 501)

**1.1.32.3.19 SUDFVolumeInfo::tRecordingDateTime Data Member new****C++**

```
SFileDateTime tRecordingDateTime;
```

**Description**

The date / time info when the disc or images was created. You can overwrite it with FileDateTimeEx.

**1.1.32.4 SUDFVolumeInfo Methods**

The methods of the SUDFVolumeInfo class are listed here.

**Public Methods**

	Name	Description
💡	INSERT_STRUCTURE_PADDING (see page 105)	This is INSERT_STRUCTURE_PADDING, a member of class SUDFVolumeInfo.

**1.1.32.4.1 SUDFVolumeInfo::INSERT\_STRUCTURE\_PADDING Method****C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_STRING_DATA);
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SUDFVolumeInfo.

**1.1.33 SVideoFormat Structure****C++**

```
struct SVideoFormat {
    BS_BOOL bUseable;
    int32 nWidth;
```

```

int32 nHeight;
int32 nBitRate;
double fFPS;
int32 nAspectRatio;
int32 nPlaytime;
};

```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK sub class that contains information of a mpeg video file.

### 1.1.33.1 SVideoFormat Members

The following tables list the members exposed by SVideoFormat.

**Public Data Members**

	Name	Description
◆	bUseable ( <a href="#">see page 107</a> )	States whether the data structure is valid and can be used.
◆	fFPS ( <a href="#">see page 107</a> )	The current frame rate in frames / second (FPS).
◆	nAspectRatio ( <a href="#">see page 107</a> )	The current aspect ratio: BS_AR_SQUARE_PIXELS ( <a href="#">see page 309</a> ), BS_AR_4TO3_DISPLAY ( <a href="#">see page 309</a> ), BS_AR_16TO9_DISPLAY ( <a href="#">see page 308</a> ), BS_AR_221TO2_DISPLAY ( <a href="#">see page 308</a> ), BS_AR_UNKNOWN ( <a href="#">see page 309</a> )
◆	nBitRate ( <a href="#">see page 107</a> )	The current bitrate of the video in bits / seconds (bits/s).
◆	nHeight ( <a href="#">see page 108</a> )	The height of the video in pixel.
◆	nPlaytime ( <a href="#">see page 108</a> )	The playtime of the video in seconds.
◆	nWidth ( <a href="#">see page 108</a> )	The width of the video in pixel.

**Public Methods**

	Name	Description
≡◆	INSERT_STRUCTURE_PADDING ( <a href="#">see page 108</a> )	This is INSERT_STRUCTURE_PADDING, a member of class SVideoFormat.
≡◆	SVideoFormat ( <a href="#">see page 106</a> )	This is SVideoFormat, a member of class SVideoFormat.

### 1.1.33.2 SVideoFormat::SVideoFormat Constructor

**C++**

```
SVideoFormat();
```

**Description**

This is SVideoFormat, a member of class SVideoFormat.

### 1.1.33.3 SVideoFormat Data Members

The data members of the SVideoFormat class are listed here.

**Public Data Members**

	Name	Description
◆	bUseable ( <a href="#">see page 107</a> )	States whether the data structure is valid and can be used.

◆	fFPS ( <a href="#">see page 107</a> )	The current frame rate in frames / second (FPS).
◆	nAspectRatio ( <a href="#">see page 107</a> )	The current aspect ratio: BS_AR_SQUARE_PIXELS ( <a href="#">see page 309</a> ), BS_AR_4TO3_DISPLAY ( <a href="#">see page 309</a> ), BS_AR_16TO9_DISPLAY ( <a href="#">see page 308</a> ), BS_AR_221TO2_DISPLAY ( <a href="#">see page 308</a> ), BS_AR_UNKNOWN ( <a href="#">see page 309</a> )
◆	nBitRate ( <a href="#">see page 107</a> )	The current bitrate of the video in bits / seconds (bits/s).
◆	nHeight ( <a href="#">see page 108</a> )	The height of the video in pixel.
◆	nPlaytime ( <a href="#">see page 108</a> )	The playtime of the video in seconds.
◆	nWidth ( <a href="#">see page 108</a> )	The width of the video in pixel.

### 1.1.33.3.1 SVideoFormat::bUseable Data Member

C++

```
BS_BOOL bUseable;
```

**Description**

States whether the data structure is valid and can be used.

### 1.1.33.3.2 SVideoFormat::fFPS Data Member

C++

```
double fFPS;
```

**Description**

The current frame rate in frames / second (FPS).

### 1.1.33.3.3 SVideoFormat::nAspectRatio Data Member

C++

```
int32 nAspectRatio;
```

**Description**

The current aspect ratio:

BS\_AR\_SQUARE\_PIXELS ([see page 309](#)),  
BS\_AR\_4TO3\_DISPLAY ([see page 309](#)),  
BS\_AR\_16TO9\_DISPLAY ([see page 308](#)),  
BS\_AR\_221TO2\_DISPLAY ([see page 308](#)),  
BS\_AR\_UNKNOWN ([see page 309](#))

### 1.1.33.3.4 SVideoFormat::nBitRate Data Member

C++

```
int32 nBitRate;
```

**Description**

The current bitrate of the video in bits / seconds (bits/s).

### 1.1.33.3.5 SVideoFormat::nHeight Data Member

**C++**

```
int32 nHeight;
```

**Description**

The height of the video in pixel.

### 1.1.33.3.6 SVideoFormat::nPlaytime Data Member

**C++**

```
int32 nPlaytime;
```

**Description**

The playtime of the video in seconds.

### 1.1.33.3.7 SVideoFormat::nWidth Data Member

**C++**

```
int32 nWidth;
```

**Description**

The width of the video in pixel.

## 1.1.33.4 SVideoFormat Methods

The methods of the SVideoFormat class are listed here.

**Public Methods**

	Name	Description
💡	INSERT_STRUCTURE_PADDING (↗ see page 108)	This is INSERT_STRUCTURE_PADDING, a member of class SVideoFormat.

### 1.1.33.4.1 SVideoFormat::INSERT\_STRUCTURE\_PADDING Method

**C++**

```
INSERT_STRUCTURE_PADDING(STRUCTURE_PADDING_NORMAL - sizeof(int32));
```

**Description**

This is INSERT\_STRUCTURE\_PADDING, a member of class SVideoFormat.

## 1.2 Functions

The following table lists functions in this documentation.

**Functions**

	Name	Description
💡	Abort (↗ see page 118)	This function interrupts the current burning process.

	AddBurnDevice ( <a href="#">see page 118</a> )	This function will add a device to the multi burn device list. This is important for multi-device burning.
	AddDir ( <a href="#">see page 118</a> )	This function adds a directory and its contents to your project.
	AddFile ( <a href="#">see page 119</a> )	This function adds a single file to your project.
	AddFileEx ( <a href="#">see page 120</a> )	This function allow you to add defined segments of a file to the project.
	AnalyseDeviceCapability ( <a href="#">see page 120</a> )	With this function you can analyze all possible capabilities of the current capability handle. You can analyze as many capabilities and as many times as needed.
	AudioFileStop ( <a href="#">see page 121</a> )	This function stops the internal audio player.
	Burn ( <a href="#">see page 121</a> )	This function writes the prepared project to the inserted disc.
	BurnDialog ( <a href="#">see page 121</a> )	This function starts the burning process with the embedded dialog.
	BurnISO ( <a href="#">see page 122</a> )	This function writes a defined ISO file to a disc.
	CheckLicenseKey ( <a href="#">see page 123</a> )	The function CheckLicenseKey validates the given key.
	CheckSignature ( <a href="#">see page 123</a> )	Function to check if the disc or image has a signature for encryption or compression.
	ClearAll ( <a href="#">see page 123</a> )	This function removes all files and folders from the current project.
	CloseCDTextHandle ( <a href="#">see page 124</a> )	This function will close a created handle to the CD-Text information of the audio disc.
	CloseDevice ( <a href="#">see page 124</a> )	This function closes the current device (tray).
	CloseDirectory ( <a href="#">see page 124</a> )	This function closes an open directory handle.
	CloseDiskSession ( <a href="#">see page 125</a> )	This function closes an open session handle.
	CloseNetworkTagsHandle ( <a href="#">see page 125</a> )	This function will close a CDDA tag handle that was initialized.
	CloseSession ( <a href="#">see page 126</a> )	This function tries to close the last open session on the current disc.
	ConvertSpeedFromKBPerSec ( <a href="#">see page 126</a> )	This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0.
	CopyDisk ( <a href="#">see page 127</a> )	This function will copy a disc directly to a selected burner.
	CreateDir ( <a href="#">see page 127</a> )	This function creates a new empty directory in the current project.
	CreateImage ( <a href="#">see page 127</a> )	This function will copy a disc to a image file.
	CreateProject ( <a href="#">see page 128</a> )	This function creates a new project. The current project will be deleted.
<b>new</b>	DABurn ( <a href="#">see page 129</a> )	This function writes the prepared project to the inserted disc. <b>Direct access version.</b>
<b>new</b>	DABurnISOImage ( <a href="#">see page 129</a> )	This function writes a defined ISO file to a disc. <b>Direct access version.</b>
<b>new</b>	DACheckSignature ( <a href="#">see page 130</a> )	Function to check if the disc or image has a signature for encryption or compression. <b>Direct access version.</b>
<b>new</b>	DACloseDevice ( <a href="#">see page 130</a> )	This function closes the current device (tray). <b>Direct access version.</b>
<b>new</b>	DACloseSession ( <a href="#">see page 130</a> )	This function tries to close the last open session on the current disc. <b>Direct access version.</b>
<b>new</b>	DACConvertSpeedFromKBPerSec ( <a href="#">see page 131</a> )	This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0. <b>Direct access version.</b>
<b>new</b>	DACopyDisk ( <a href="#">see page 131</a> )	This function will copy a disc directly to a selected burner. <b>Direct access version.</b>

 <a href="#">DACreateImage</a>	( <a href="#">see page 132</a> )	This function will copy a disc to a image file. <b>Direct access version.</b>
 <a href="#">DAEjectDevice</a>	( <a href="#">see page 132</a> )	This function opens the current device (tray). <b>Direct access version.</b>
 <a href="#">DAErase</a>	( <a href="#">see page 133</a> )	This function erases the disc in the current device. <b>Direct access version.</b>
 <a href="#">DAGetBurnSpeed</a>	( <a href="#">see page 133</a> )	This function returns the writing speed of the selected device. <b>Direct access version.</b>
 <a href="#">DAGetDeviceCapabilities</a>	( <a href="#">see page 134</a> )	If you need to know the device capabilities use this function. <b>Direct access version.</b> <b>DEPRECATED! IS RELPACED BY:</b> <a href="#">AnalyseDeviceCapability</a> ( <a href="#">see page 120</a> ())
 <a href="#">DAGetDeviceCapabilitiesHandle</a>	( <a href="#">see page 134</a> )	This function will provide a capabilite handle accoring to the gives device index. Call this function before you call <a href="#">AnalyseDeviceCapability</a> ( <a href="#">see page 120</a> ()). <b>Direct access version.</b>
 <a href="#">DAGetDeviceInformation</a>	( <a href="#">see page 135</a> )	This function reads the device information of a device from the internal device list. <b>Direct access version.</b>
 <a href="#">DAGetDeviceInformationEx</a>	( <a href="#">see page 135</a> )	This function reads the device information of a device from the internal device list. <b>Direct access version.</b>
 <a href="#">DAGetMaxBurnSpeed</a>	( <a href="#">see page 136</a> )	This function returns the max. writing speed of the current device. <b>Direct access version.</b>
 <a href="#">DAGetMaxReadSpeed</a>	( <a href="#">see page 136</a> )	This function reads the max. reading speed of the current device. <b>Direct access version.</b>
 <a href="#">DAGetMediumFreedBld</a>	( <a href="#">see page 137</a> )	Use this function will receive the unique disc id for the audio disc. <b>Direct access version.</b>
 <a href="#">DAGetMediumInformation</a>	( <a href="#">see page 137</a> )	This function returns the information about the current disc in the selected device. <b>Direct access version.</b>
 <a href="#">DAGetPossibleBurnSpeeds</a>	( <a href="#">see page 138</a> )	This function returns a structure of the available burning speeds of the current device. <b>Direct access version.</b>
 <a href="#">DAGetPossibleImageFormats</a>	( <a href="#">see page 138</a> )	This function will check what kind of image formats are possible to save from selected disc. <b>Direct access version.</b>
 <a href="#">DAGetPossibleReadSpeeds</a>	( <a href="#">see page 139</a> )	This function returns a structure of the available reading speeds of the current device. <b>Direct access version.</b>
 <a href="#">DAGetReadSpeed</a>	( <a href="#">see page 139</a> )	This function returns the reading speed of the current device. <b>Direct access version.</b>
 <a href="#">DAGetSessionInformation</a>	( <a href="#">see page 140</a> )	This function returns the session information of the current disc in the current device. <b>Direct access version.</b>
 <a href="#">DAGetTrackFormatEx</a>	( <a href="#">see page 140</a> )	This function returns the type of the disc track of the current medium of the given device. <b>Direct access version.</b>
 <a href="#">DAGetTrackIndexes</a>	( <a href="#">see page 141</a> )	This function returns a list of sub indexes of the give audio track. <b>Direct access version.</b>

 DAGetTrackInformation (see page 141)	This function returns the track information of the current disc in the current device. <b>Direct access version.</b>
 DAGetTrackISRC (see page 142)	This function returns the ISRC code from the give audio track. <b>Direct access version.</b>
 DAGrabAudioTrack (see page 142)	The function will grab/save a audio track to a file. Tagging and different encoding types are available. <b>Direct access version.</b>
 DAImportFile (see page 143)	This function imports a file from a session to your project. <b>Direct access version.</b>
 DAImportFileEx (see page 143)	This function imports a file from a session to your project. <b>Direct access version.</b>
 DAIsDeviceReady (see page 144)	This function tests if the current device is ready. <b>Direct access version.</b>
 DALockMedium (see page 144)	This function will lock/unlock a disc inside a selected device. <b>Direct access version.</b>
 DAOpenDiskSession (see page 145)	This function opens a session on the selected disc/device. <b>Direct access version.</b>
 DAPlayAudioTrack (see page 145)	This function will play a audio track on the current disc or disc image. <b>Direct access version.</b>
 DAPrepare (see page 146)	This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables. <b>Direct access version.</b>
 DAReadCDText (see page 146)	This function will create a handle to the CD-Text information of the audio disc according to the give device index. <b>Direct access version.</b>
 DAReadFileContents (see page 147)	This function reads a segment of file from disc to a buffer. <b>Direct access version.</b>
 DAReadSectors (see page 147)	This function will read a given sector into a buffer. <b>Direct access version.</b>
 DASaveTrackToFile (see page 148)	This function saves a selected track to a file. <b>Direct access version.</b>
 DASetBurnSpeed (see page 148)	This function sets the writing speed of the current device. <b>Direct access version.</b>
 DASetReadSpeed (see page 149)	This function sets the reading speed of the current device. <b>Direct access version.</b>
 DASetRegionalCode (see page 149)	This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0. <b>Direct access version.</b>
 DAVerifyFile (see page 150)	This function verify a file of the medium against whether it is readable without any error. <b>Direct access version.</b>
 DelInitALL (see page 150)	This is function DelInitALL.
 DelInitialize (see page 151)	This function de-initializes the IsoSDK. Components of the IsoSDK will be released from memory.
 DeleteProject (see page 151)	This function deletes the current project.
 DirExists (see page 151)	This function will check whether a directory exists inside the project tree.
 EjectDevice (see page 152)	This function opens the current device (tray).
 EnableImageDevice (see page 152)	Enable or disable the internal ImageWriter device inside the device list.

	EnableMCNDisabling ( <a href="#">see page 153</a> )	This function will enable / disable the media change notification event on windows driven computers. Use this function to work with robotic systems.
	Erase ( <a href="#">see page 153</a> )	This function erases the disc in the current device.
	EraseDialog ( <a href="#">see page 154</a> )	This function starts the deletion process with the embedded dialog.
	EraseMpegByIndex ( <a href="#">see page 154</a> )	This function deletes a previously added MPEG file from a VCD or SVCD project.
	ForceDeInitialize ( <a href="#">see page 154</a> )	This is function ForceDeInitialize.
	GetActiveDevicesCount ( <a href="#">see page 155</a> )	This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.
	GetASPI ( <a href="#">see page 155</a> )	With this function you can get the information about the currently used ASPI layer.
	GetAudioFileSize ( <a href="#">see page 156</a> )	Use this function enumerate the final size of the compressed audio file
	GetAudioTags ( <a href="#">see page 156</a> )	This is function GetAudioTags.
	GetBootInfoEx ( <a href="#">see page 156</a> )	Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.
	GetBootVolumeInformation ( <a href="#">see page 157</a> )	Use this function to receive information about the boot file system on a existing disc or image.
	GetBurnDevice ( <a href="#">see page 157</a> )	This function returns the currently selected burn device. If you use multiple devices this function will return more than one device. Check the array with GetActiveDeviceCount()
	GetBurnDevices ( <a href="#">see page 158</a> )	This function returns an array with all selected burn devices.
	GetBurnDoneEventCallback ( <a href="#">see page 158</a> )	Hooks an already set BurnDoneEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetBurnFileEventCallback ( <a href="#">see page 159</a> )	Hooks an already set BurnFileEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetBurnSpeed ( <a href="#">see page 159</a> )	This function returns the writing speed of the selected device.
	GetCDTextDiskTagString ( <a href="#">see page 160</a> )	This function will get a specified CD-Text item from the created CD-Text handle of a audio disc.
	GetCDTextTrackTagString ( <a href="#">see page 160</a> )	This function will get a specified CD-Text track item from the created CD-Text handle of a audio disc.
	GetCompareFilesForArrangementEventCallback ( <a href="#">see page 161</a> )	Hooks an already set CompareFilesForArrangementEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetCompressEncrypt ( <a href="#">see page 161</a> )	This function will get the parameters for internal compression and encryption.
	GetDeviceCapabilities ( <a href="#">see page 162</a> )	If you need to know the device capabilities use this function. <b>DEPRECATED! IS RELPACED BY:</b> AnalyseDeviceCapability ( <a href="#">see page 120</a> )()
	GetDeviceCapabilitiesHandle ( <a href="#">see page 162</a> )	This function will provide a capabilite handle accoring to the gives device index. Call this function before you call AnalyseDeviceCapability ( <a href="#">see page 120</a> )().
	GetDeviceInformation ( <a href="#">see page 163</a> )	This function reads the device information of a device from the internal device list.
	GetDeviceInformationEx ( <a href="#">see page 163</a> )	This function reads the device information of a device from the internal device list.
	GetDevices ( <a href="#">see page 164</a> )	This function returns a list of all available devices or a list of available devices with burning capabilities.
	GetDVDVideoOptions ( <a href="#">see page 164</a> )	This function gets the special options for Video DVD projects.
	GetEraseDoneEventCallback ( <a href="#">see page 165</a> )	Hooks an already set EraseDoneEvent ( <a href="#">see page 238</a> ). Use this if you need the event in another dialog
	GetErrorDeviceName ( <a href="#">see page 165</a> )	This function will return the name of the drive that reported the last error. This is helpful if you use multi-device burning.

	GetFileAllocationTable ( <a href="#">see page 165</a> )	Use this function to receive information of the file allocation table and the file extents on the disc according to the given file.
	GetFileEntry ( <a href="#">see page 166</a> )	This function will allow you to get all information about an already added file.
	GetFileTimeEx ( <a href="#">see page 166</a> )	Use this function to read the extended FileTimeEx data for the project types.
	GetFinalizeEventCallback ( <a href="#">see page 167</a> )	Hooks an already set FinalizeEvent ( <a href="#">see page 239</a> ). Use this if you need the event in another dialog.
	GetImagePath ( <a href="#">see page 167</a> )	This function will received the path to a opened image file.
	GetImageSize ( <a href="#">see page 168</a> )	This function returns the project size. The GetImageSize function is only available after the execution of the Prepare ( <a href="#">see page 194</a> ) function.
	GetInfoTextEventCallback ( <a href="#">see page 168</a> )	Hooks an already set InfoTextEvent ( <a href="#">see page 240</a> ). Use this if you need the event in another dialog.
	GetISOInfoEx ( <a href="#">see page 168</a> )	Use this function to read the extended ISO fields for Data CD/DVD/BD and ISO images. Available only in data projects.
	GetISOVolumeInformation ( <a href="#">see page 169</a> )	Use this function to receive information about the ISO filesystem on a existing disc or image.
	GetJobDoneEventCallback ( <a href="#">see page 169</a> )	Hooks an already set JobDoneEvent ( <a href="#">see page 241</a> ). Use this if you need the event in another dialog.
	GetLanguage ( <a href="#">see page 170</a> )	Use this function to determine the current language of the IsoSDK.
	GetMaxBurnSpeed ( <a href="#">see page 170</a> )	This function returns the max. writing speed of the current device.
	GetMaxReadSpeed ( <a href="#">see page 171</a> )	This function reads the max. reading speed of the current device.
	GetMediumFreedId ( <a href="#">see page 171</a> )	Use this function will receive the unique disc id for the audio disc.
	GetMediumInformation ( <a href="#">see page 172</a> )	This function returns the information about the current disc in the selected device.
	GetMpegCount ( <a href="#">see page 172</a> )	This function returns the number of added files from a VCD or SVCD project.
	GetNetworkDiskTagInt ( <a href="#">see page 172</a> )	This function will received a numeric value of a Cddb disc tag.
	GetNetworkDiskTagString ( <a href="#">see page 173</a> )	This function will received a string value of a Cddb disc tag.
	GetNetworkTrackTagInt ( <a href="#">see page 174</a> )	This function will received a numeric value of a Cddb track tag according to the given track.
	GetNetworkTrackTagString ( <a href="#">see page 174</a> )	This function will received a string value of a Cddb track tag according to the given track.
	GetNumberOfFiles ( <a href="#">see page 175</a> )	This function returns the number of files of the current disc in the current device.
	GetOptions ( <a href="#">see page 175</a> )	Use this function to read the burning options.
	GetPlayTime ( <a href="#">see page 176</a> )	This function returns the playtime of the given audio file in seconds.
	GetPossibleBurnSpeeds ( <a href="#">see page 176</a> )	This function returns a structure of the available burning speeds of the current device.
	GetPossibleImageFormats ( <a href="#">see page 177</a> )	This function will check what kind of image formats are possible to save from selected disc.
	GetPossibleReadSpeeds ( <a href="#">see page 177</a> )	This function returns a structure of the available reading speeds of the current device.
	GetPrecisePlayTime ( <a href="#">see page 178</a> )	This function returns the playtime of the given audio file in milliseconds.
	GetProcessEventCallback ( <a href="#">see page 178</a> )	Hooks an already set ProcessEvent ( <a href="#">see page 242</a> ). Use this if you need the event in another dialog.

	GetProjectType (see page 178)	This function returns the type of the current project.
	GetRAWDataEventCallback (see page 179)	Hooks an already set RAWDataEvent (see page 242). Use this if you need the event in another dialog.
	GetReadDevice (see page 179)	This function returns the currently selected read device.
	GetReadSpeed (see page 180)	This function returns the reading speed of the current device.
	GetSessionInformation (see page 180)	This function returns the session information of the current disc in the current device.
	GetStartVerifyEventCallback (see page 181)	Hooks an already set StartVerifyEvent (see page 243). Use this if you need the event in another dialog.
	GetText (see page 181)	This function returns the text of a text ID according to your language setting; e.g. error code.
	GetTmpPath (see page 182)	To determine the current temporary path, call this function.
	GetTrackFormatEx (see page 182)	This function returns the type of the disc track of the current medium of the given device.
	GetTrackIndexes (see page 183)	This function returns a list of sub indexes of the give audio track.
	GetTrackInformation (see page 183)	This function returns the track information of the current disc in the current device.
	GetTrackISRC (see page 184)	This function returns the ISRC code from the give audio track.
	GetUDFOptions (see page 184)	Use this function to set the UDF options for UDF data projects.
	GetUDFOptionsEx (see page 185)	Use this function to read the UDF extended options for UDF data projects.
	GetUDFVolumeInformation (see page 185)	Use this function to receive information about the UDF file system on a existing disc or image.
	GetVerify (see page 185)	Use this function to get the afterburn verification state.
	GetVerifyDoneEventCallback (see page 186)	Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.
	GetVerifyErrorEventCallback (see page 186)	Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.
	GetVerifyFileEventCallback (see page 187)	Hooks an already set VerifyFileEvent (see page 245). Use this if you need the event in another dialog.
	GetVerifySectorEventCallback (see page 187)	Hooks an already set VerifySectorEvent (see page 245). Use this if you need the event in another dialog.
	GetWriteCDTextInUnicode (see page 187)	Use this function to read the CD-Text Unicode switch.
	GrabAudioTrack (see page 188)	The function will grab/save a audio track to a file. Tagging and different encoding types are available.
	ImportFile (see page 188)	This function imports a file from a session to your project.
	ImportFileEx (see page 189)	This function imports a file from a session to your project.
	Initialize (see page 189)	This function initializes the IsoSDK. The function must be called before you can use other functions.
	IsDeviceReady (see page 190)	This function tests if the current device is ready.
	IsValidVideoTsFolder (see page 190)	This function will check if a given path contains a valid VideoDVD structure with IFO, BUP and VOB Files.
	LoadBassPlugin (see page 191)	This function will load a given Bass plugin to the IsoSDK. Bass.dll have to be present. <a href="http://www.un4seen.com/">http://www.un4seen.com/</a> <b>NOTE:</b> The bass.dll and plugins are available as 32 and 64 bit.
	LockMedium (see page 191)	This function will lock/unlock a disc inside a selected device.
	MultiDeviceDialog (see page 192)	This function starts the multi-device selection dialog to burn on more than one device.
	OpenDirectory (see page 192)	This function opens a directory on the selected session.

	OpenDiskSession ( <a href="#">see page 193</a> )	This function opens a session on the selected disc/device.
	PlayAudioFile ( <a href="#">see page 193</a> )	Use this function to play the given file to the default output device.
	PlayAudioTrack ( <a href="#">see page 194</a> )	This function will play a audio track on the current disc or disc image.
	Prepare ( <a href="#">see page 194</a> )	This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables.
	ReadCDText ( <a href="#">see page 194</a> )	This function will create a handle to the CD-Text information of the audio disc according to the give device index.
	ReadDirectory ( <a href="#">see page 195</a> )	This function returns a pointer to a structure representing the directory entry at the specified position.
	ReadFileContents ( <a href="#">see page 195</a> )	This function reads a segment of file from disc to a buffer.
	ReadSectors ( <a href="#">see page 196</a> )	This function will read a given sector into a buffer.
	ReleaseDeviceCapabilities ( <a href="#">see page 196</a> )	Release a created capabilities handle to clean up the memory.
	RemoveBurnDevice ( <a href="#">see page 197</a> )	This function will remove a device from the multi burn device list. This is important for multi-device burning.
	RemoveDir ( <a href="#">see page 197</a> )	This function removes a directory from the current project.
	RemoveFile ( <a href="#">see page 198</a> )	
	RenameDir ( <a href="#">see page 198</a> )	This function rename / move an directory in the project.
	RenameFile ( <a href="#">see page 198</a> )	This function rename / move a file in the project.
	RescanDevices ( <a href="#">see page 199</a> )	This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.
	ResetCallbacks ( <a href="#">see page 199</a> )	This function will reset all initialized callbacks to your project.
	SaveLogToFile ( <a href="#">see page 200</a> )	This function will save the automatically generated burning log data to a file.
	SaveOptionsToFile ( <a href="#">see page 200</a> )	This function will save the burn settings/ options to a XML File.
	SaveTrackToFile ( <a href="#">see page 201</a> )	This function saves a selected track to a file.
	SetAddFileEventCallback ( <a href="#">see page 201</a> )	Specifies an own function as a callback of the type AddFileEvent ( <a href="#">see page 235</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetASPI ( <a href="#">see page 202</a> )	If you want to use the internal ASPI layer after the initialization you use this function to switch from the external to the internal ASPI layer et vice versa .
	SetAudioDecodeDoneEventCallback ( <a href="#">see page 203</a> )	Specifies an own function as a callback of the type AudioDecodeDoneEvent ( <a href="#">see page 236</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetAudioDecoderEventCallback ( <a href="#">see page 203</a> )	Specifies an own function as a callback of the type AudioDecoderEvent ( <a href="#">see page 236</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetAudioFileProperty ( <a href="#">see page 204</a> )	The function sets the audio properties for the specified audio file in the project or for the disc. Supported projects: Audio, Mixed Mode.
	SetBootInfoEx ( <a href="#">see page 204</a> )	Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.
	SetBurnDevice ( <a href="#">see page 205</a> )	This function sets the current device. This drive will be used for the burning process.
	SetBurnDoneEventCallback ( <a href="#">see page 205</a> )	Specifies an own function as a callback of the type BurnDoneEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetBurnFileEventCallback ( <a href="#">see page 206</a> )	Specifies an own function as a callback of the type BurnFileEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetBurnSpeed ( <a href="#">see page 206</a> )	This function sets the writing speed of the current device.

	SetCompareFilesForArrangementEventCallback ( <a href="#">see page 207</a> )	Specifies an own function as a callback of the type CompareFilesForArrangementEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetCompressEncrypt ( <a href="#">see page 208</a> )	This function will set the parameters for internal compression and encryption.
	SetCreateDirEventCallback ( <a href="#">see page 208</a> )	Specifies an own function as a callback of the type CreateDirEvent ( <a href="#">see page 238</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetDVDVideoOptions ( <a href="#">see page 209</a> )	This function sets the special options for VideoDVD projects.
	SetEraseDoneEventCallback ( <a href="#">see page 209</a> )	Specifies an own function as a callback of the type EraseDoneEvent ( <a href="#">see page 238</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetFileAttr ( <a href="#">see page 210</a> )	This function set the attributes to a file.
	SetFileTimeEx ( <a href="#">see page 210</a> )	<p>This function is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.</p> <p>You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes (<a href="#">see page 211</a>) method.</p> <p>This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.</p>
	SetFileTimes ( <a href="#">see page 211</a> )	This function will set the custom dates to a file.
	SetFileUserParam ( <a href="#">see page 211</a> )	This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.
	SetFinalizeEventCallback ( <a href="#">see page 212</a> )	Specifies an own function as a callback of the type FinalizeEvent ( <a href="#">see page 239</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetFXApp ( <a href="#">see page 212</a> )	Set the Fox-Toolkit App handle to the SDK.
	SetGetTextEventCallback ( <a href="#">see page 213</a> )	Specifies an own function as a callback of the type GetTextEvent ( <a href="#">see page 239</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetIgnoreFileExist ( <a href="#">see page 213</a> )	The IsoSDK will check about already added files while AddFile ( <a href="#">see page 119</a> ) and AddDir ( <a href="#">see page 118</a> ) function. If you have industrial workflows with thousands of files you can disable this check with this function.
	SetImagePath ( <a href="#">see page 214</a> )	The function sets the path to the image (*.iso) file you want to burn using the 'Imagewriter'.
	SetInfoTextEventCallback ( <a href="#">see page 214</a> )	Specifies an own function as a callback of the type InfoTextEvent ( <a href="#">see page 240</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetISOInfoEx ( <a href="#">see page 215</a> )	Use this function to set the extended ISO Fields for Data CD/DVD and ISO images. Available only in data projects.
	SetJobDoneEventCallback ( <a href="#">see page 215</a> )	Specifies an own function as a callback of the type JobDoneEvent ( <a href="#">see page 241</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetLanguage ( <a href="#">see page 216</a> )	This function sets the language of the IsoSDK. The following resources are effected: error codes, message codes and GUI codes.
	SetOptions ( <a href="#">see page 217</a> )	Use this function to set the burning options.
	SetOptionsFromFile ( <a href="#">see page 217</a> )	This function will set the burn settings/ options from a previously stored XML file.
	SetProcessEventCallback ( <a href="#">see page 217</a> )	Specifies an own function as a callback of the type ProcessEvent ( <a href="#">see page 242</a> ). This function will be called every time the event triggers inside the IsoSDK.

	SetRAWDataEventCallback ( <a href="#">see page 218</a> )	Specifies an own function as a callback of the type RAWDataEvent ( <a href="#">see page 242</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetRAWStructure ( <a href="#">see page 219</a> )	The function sets the structure of the disc and the format of the input data for a RAW project.
	SetReadDevice ( <a href="#">see page 219</a> )	This function sets the reading speed of the current read device.
	SetReadSpeed ( <a href="#">see page 219</a> )	This function sets the reading speed of the current device.
	SetRegionalCode ( <a href="#">see page 220</a> )	This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0.
	SetRemoveFileEventCallback ( <a href="#">see page 221</a> )	Specifies an own function as a callback of the type RemoveFileEvent ( <a href="#">see page 243</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetStartVerifyEventCallback ( <a href="#">see page 221</a> )	Specifies an own function as a callback of the type StartVerifyEvent ( <a href="#">see page 243</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetTmpPath ( <a href="#">see page 222</a> )	With the help of this function the temporary folder is set. A previously set path will be overwritten.
	SetUDFOptions ( <a href="#">see page 222</a> )	Use this function to set the UDF options for UDF data projects.
	SetUDFOptionsEx ( <a href="#">see page 223</a> )	Use this function to set the UDF extended options for UDF data projects. Insert additional UDF settings with this function.
	SetVCDKeyHandler ( <a href="#">see page 223</a> )	This function sets which item will be shown next when particular key is pressed.
	SetVCDTimeOutHandler ( <a href="#">see page 224</a> )	This function sets transition timeout between items.
	SetVerify ( <a href="#">see page 224</a> )	Use this function to set the after burn verification.
	SetVerifyDoneEventCallback ( <a href="#">see page 224</a> )	Specifies an own function as a callback of the type VerifyDoneEvent ( <a href="#">see page 244</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVerifyErrorEventCallback ( <a href="#">see page 225</a> )	Specifies an own function as a callback of the type VerifyErrorEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the SDK.
	SetVerifyFileEventCallback ( <a href="#">see page 226</a> )	Specifies an own function as a callback of the type VerifyFileEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVerifySectorEventCallback ( <a href="#">see page 226</a> )	Specifies an own function as a callback of the type VerifySectorEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVideoScanDoneEventCallback ( <a href="#">see page 227</a> )	Specifies an own function as a callback of the type VideoScanDoneEvent ( <a href="#">see page 246</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVideoScannerEventCallback ( <a href="#">see page 228</a> )	Specifies an own function as a callback of the type VideoScannerEvent ( <a href="#">see page 246</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetWriteCDTextInUnicode ( <a href="#">see page 228</a> )	Use this function to set the CD-Text Unicode switch.
	StopMpegAction ( <a href="#">see page 229</a> )	This function interupts the current scanning process of an added video file to a VCD or SVCD project.
	TagsFromNetworkDialog ( <a href="#">see page 229</a> )	This function will create a handle to the CDDDB information of the audio disc according to the give device index.
	VerifyFile ( <a href="#">see page 230</a> )	This function verify a file of the medium against whether it is readable without any error.

## 1.2.1 Abort Function

### C++

```
int32 BS_CALL Abort();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function interrupts the current burning process.

### Notes

Depending on the system or device, the abort process may take some time to be completed.

## 1.2.2 AddBurnDevice Function

### C++

```
int32 BS_CALL AddBurnDevice(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR* lpszDevice	The device/drive letter of the device to be added.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will add a device to the multi burn device list. This is important for multi-device burning.

### Notes

In compare to previous versions, the IsoSDK do not only use the first letter or number of the device string.

## 1.2.3 AddDir Function

### C++

```
int32 BS_CALL AddDir(struct SDirToAdd DirToAdd);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
struct SDirToAdd DirToAdd	A structure SDirToAdd (see page 16).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function adds a directory and its contents to your project.

**Notes**

This function adds a directory or folder to a project with all files and subfolders. To create an empty folder use CreateDir (see page 127)().

With SetIgnoreNotExist (see page 213) you can speed up this function if you want to add many files or directory to the project.

## 1.2.4 AddFile Function

**C++**

```
int32 BS_CALL AddFile(struct SFileToAdd FileToAdd);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SFileToAdd FileToAdd	A structure SFileToAdd (see page 47).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function adds a single file to your project.

**Notes**

- Audio burning project: the file must either be a valid mp3, OGG or WAV file with 2 audio channels, 44.100 KHz and 16 Bit.
- Cue burning project: the file must be a valid cue file with corresponding bin file.
- VCD burning project: the file must be an ISO compatible MPEG1 file.
- SVCD burning project: the file must be an ISO compatible MPEG2 file.
- The value lpszDestinationPath is based on the rootDirectory. The rootDirectory is always "\\\" expected MixedMode project. Here the root for audio is "\\audio" and for data files "\\data".
- With SetIgnoreNotExist (see page 213) you can speed up this function if you want to add many files or directory to the project.
- If ISO filesystem is used and not ISOLevel 3 and the file size is up 2 GB limitation, this function will throw an error.

## 1.2.5 AddFileEx Function

### C++

```
int32 BS_CALL AddFileEx(struct SFileToAddEx FileToAdd);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SFileToAddEx FileToAdd	A structure SFileToAddEx ( <a href="#">see page 50</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function allow you to add defined segments of a file to the project.

### Notes

Only allowed in data projects.

## 1.2.6 AnalyseDeviceCapability Function

### C++

```
int32 BS_CALL AnalyseDeviceCapability(int32 CapabilitiesHandle, int64 nCapabilityName,
BS_BOOL * pbResult);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 CapabilitiesHandle	A handle that was created with GetDeviceCapabilitiesHandle ( <a href="#">see page 162</a> ()).
int64 nCapabilityName	The capabilitie to check.
BS_BOOL * pbResult	A pointer to a BS_BOOL ( <a href="#">see page 312</a> ) variable to receive the state.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

With this function you can analyze all possible capabilities of the current capabilitie handle. You can analyze as many capabilities and as many times as needed.

### Example

```
//Create a handle
int32 nCapsHandle;
//Set the handle
int32 nErr;
```

```

nErr = GetDeviceCapabilitiesHandle(BS_CURRENT_DEVICE, &nCapsHandle);
//Analyze a capability
BS_BOOL bDeviceReadsCD_RWs;
nErr = ::AnalyseDeviceCapability(nCapsHandle, BS_READ_CDRW, &bDeviceReadsCD_RWs);
//Release the handle
nErr = ReleaseDeviceCapabilities(nCapsHandle);

```

## 1.2.7 AudioFileStop Function

### C++

```
int32 BS_CALL AudioFilestop();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function stops the internal audio player.

## 1.2.8 Burn Function

### C++

```
int32 BS_CALL Burn();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function writes the prepared project to the inserted disc.

### Notes

Before you can use this method, the project must have been prepared for burning with the help of the Prepare ([see page 194](#)) function.

## 1.2.9 BurnDialog Function

### C++

```
int32 BS_CALL BurnDialog(BS_BOOL bAutoRun, BS_BOOL bOpenSettings, BS_BOOL bViewSettings,
const TCHAR * pchTitle, const TCHAR * pchIsoFilePath);
```

### File

IsoSDKExport.h ([see page 577](#))

## Parameters

Parameters	Description
BS_BOOL bAutoRun	If you want to start the burning process immediately after opening the window you can set the value to BS_TRUE (see page 499). Otherwise, set it to BS_FALSE (see page 335).
BS_BOOL bOpenSettings	If you want to show the extended settings when the dialog opens, pass BS_TRUE (see page 499). BS_FALSE (see page 335) (default) means that the settings are not displayed, but they still can be opened by the user (see ViewSettings).
BS_BOOL bViewSettings	If you want to show the settings of the dialog, pass BS_TRUE (see page 499). If not, pass BS_FALSE (see page 335).
const TCHAR * pchTitle	With this value you can display your own window title in the dialog. Pass NULL, if you do not want to use this feature.
const TCHAR * pchIsoFilePath	If you want to write a selected ISO file instead of the project, pass the file name of the ISO file.

## Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

## Description

This function starts the burning process with the embedded dialog.

## Notes

This function is only available inside the FoxToolkit libraries. This function use the XServer functionality.

## 1.2.10 BurnISO Function

### C++

```
int32 BS_CALL BurnISO(const TCHAR * pchFilePath, struct SOptions options);
```

### File

IsoSDKExport.h (see page 577)

### Parameters

Parameters	Description
const TCHAR * pchFilePath	The path and name of the ISO file to be written.
struct SOptions options	A structure SOptions (see page 75).

## Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

## Description

This function writes a defined ISO file to a disc.

## Notes

The IsoSDK was designed to write ISO compatible files. If the selected ISO file was not built with the IsoSDK, problems might occur.

## 1.2.11 CheckLicenseKey Function

### C++

```
int32 BS_CALL CheckLicenseKey(const TCHAR chLicenseKey[35]);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<code>const TCHAR chLicenseKey[35]</code>	A string that contains the license key that was submitted after your purchase.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

The function CheckLicenseKey validates the given key.

## 1.2.12 CheckSignature Function

### C++

```
int32 BS_CALL CheckSignature();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Function to check if the disc or image has a signature for encryption or compression.

## 1.2.13 ClearAll Function

### C++

```
int32 BS_CALL ClearAll();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function removes all files and folders from the current project.

## 1.2.14 CloseCDTextHandle Function

### C++

```
int32 BS_CALL CloseCDTextHandle(int32 hOpenedHandle);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 hOpenedHandle	The handle that was created with ReadCDText ( <a href="#">see page 194</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will close a created handle to the CD-Text information of the audio disc.

---

## 1.2.15 CloseDevice Function

### C++

```
int32 BS_CALL CloseDevice(int32 index);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function closes the current device (tray).

---

## 1.2.16 CloseDirectory Function

### C++

```
int32 BS_CALL CloseDirectory(HDIR hDirectory);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
HDIR hDirectory	Handle to an open directory.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function closes an open directory handle.

## 1.2.17 CloseDiskSession Function

**C++**

```
int32 BS_CALL CloseDiskSession(HSESSION hSession);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
HSESSION hSession	Handle to an open session.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function closes an open session handle.

## 1.2.18 CloseNetworkTagsHandle Function

**C++**

```
int32 BS_CALL CloseNetworkTagsHandle(int32 hOpenedHandle);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 hOpenedHandle	The handle that was created with TagsFromNetworkDialog ( <a href="#">see page 229</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will close a CDDB tag handle that was initialized.

## 1.2.19 CloseSession Function

### C++

```
int32 BS_CALL CloseSession(int32 index);
```

### File

IsoSDKEExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function tries to close the last open session on the current disc.

### Notes

Open sessions can occur if a burning process fails. This function tries to close the session. However, this fails in some cases.

## 1.2.20 ConvertSpeedFromKBPerSec Function

### C++

```
int32 BS_CALL ConvertSpeedFromKBPerSec(int32 nSpeedInKBPerSec, float * pfConvertedSpeed);
```

### File

IsoSDKEExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 nSpeedInKBPerSec	The speed in kb/s you need to convert to readable speed. Like 7200 to get 48.0.
float * pfConvertedSpeed	Pointer to the readable speed information like 48.0.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0.

### Notes

You can use this function for internal calculations.

## 1.2.21 CopyDisk Function

### C++

```
int32 BS_CALL CopyDisk(SDiskCopyOptions cDiskCopyOptions);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
SDiskCopyOptions cDiskCopyOptions	A structure SDiskCopyOptions ( <a href="#">see page 24</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will copy a disc directly to a selected burner.

### Notes

Make sure that you call SetReadDevice ([see page 219](#)()) and SetBurnDevice ([see page 205](#)()) before you call this function.

## 1.2.22 CreateDir Function

### C++

```
int32 BS_CALL CreateDir(struct SDirToCreate DirToDelete);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SDirToCreate DirToDelete	A structure SDirToCreate ( <a href="#">see page 20</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function creates a new empty directory in the current project.

## 1.2.23 CreateImage Function

### C++

```
int32 BS_CALL CreateImage(SCreateImageParams cCreateImageParams, int8 nTaskType);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
SCreateImageParams cCreateImageParams	A structure SCreateImageParams ( <a href="#">see page 14</a> ).
int8 nTaskType	The type of operation. Possible values are: BS_IMGTASK_CREATE ( <a href="#">see page 358</a> ) BS_IMGTASK_VERIFY ( <a href="#">see page 358</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will copy a disc to a image file.

## 1.2.24 CreateProject Function

**C++**

```
int32 BS_CALL CreateProject(int32 nType, int32 nSessionToContinue);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
int32 nType	This is the project type. The following values are valid: BS_BURNER_AUDIO ( <a href="#">see page 314</a> ) BS_BURNER_CUE ( <a href="#">see page 314</a> ) BS_BURNER_DATA ( <a href="#">see page 315</a> ) BS_BURNER_VCD ( <a href="#">see page 316</a> ) BS_BURNER_SVCD ( <a href="#">see page 316</a> ) BS_BURNER_VIDEODVD ( <a href="#">see page 316</a> ) BS_BURNER_UDFDVD ( <a href="#">see page 316</a> ) BS_BURNER_ISOUDF ( <a href="#">see page 315</a> ) BS_BURNER_BLURAY ( <a href="#">see page 314</a> ) BS_BURNER_MIXEDMODE ( <a href="#">see page 315</a> ) BS_BURNER_RAW ( <a href="#">see page 315</a> )
int32 nSessionToContinue	The session that will be imported to the new session. Use -1 to import no session or 0 to import the last session. This parameter is only valid for data projects and will be ignored in other projects.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function creates a new project. The current project will be deleted.

**Notes**

During the initialization the IsoSDK creates a data burning project by default.

## 1.2.25 DABurn Function [new](#)

### C++

```
int32 BS_CALL DABurn(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>const</b> TCHAR* <b>lpszDevice</b>	The name of the device to be selected.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function writes the prepared project to the inserted disc.

**Direct access version.**

### Notes

Before you can use this method, the project must have been prepared for burning with the help of the Prepare ([see page 194](#)) function.

---

## 1.2.26 DABurnISOImage Function [new](#)

### C++

```
int32 BS_CALL DABurnISOImage(const TCHAR* lpszDevice, const TCHAR* pchFilePath, struct SOptions options);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>const</b> TCHAR* <b>lpszDevice</b>	The name of the device to be selected.
<b>const</b> TCHAR* <b>pchFilePath</b>	The path and name of the ISO file to be written.
<b>struct</b> SOptions <b>options</b>	A structure SOptions ( <a href="#">see page 75</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function writes a defined ISO file to a disc.

**Direct access version.**

### Notes

The IsoSDK was designed to write ISO compatible files. If the selected ISO file was not built with the IsoSDK, problems might occur.

## 1.2.27 DACheckSignature Function [new](#)

### C++

```
int32 BS_CALL DACheckSignature(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<code>const TCHAR* lpszDevice</code>	The name of the device to be selected.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Function to check if the disc or image has a signature for encryption or compression.

**Direct access version.**

---

## 1.2.28 DACloseDevice Function [new](#)

### C++

```
int32 BS_CALL DACloseDevice(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<code>const TCHAR* lpszDevice</code>	The name of the device to be selected.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function closes the current device (tray).

**Direct access version.**

---

## 1.2.29 DACloseSession Function [new](#)

### C++

```
int32 BS_CALL DACloseSession(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function tries to close the last open session on the current disc.

**Direct access version.****Notes**

Open sessions can occur if a burning process fails. This function tries to close the session. However, this fails in some cases.

## 1.2.30 DAConvertSpeedFromKBPerSec Function new

**C++**

```
int32 BS_CALL DAConvertSpeedFromKBPerSec(const TCHAR* lpszDevice, int32 nSpeedInKBPerSec,
                                         float* pfConvertedSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nSpeedInKBPerSec	The speed in kb/s you need to convert to readable speed. Like 7200 to get 48.0.
float* pfConvertedSpeed	Pointer to the readable speed information like 48.0.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0.

**Direct access version.****Notes**

You can use this function for internal calculations.

## 1.2.31 DACopyDisk Function new

**C++**

```
int32 BS_CALL DACopyDisk(const TCHAR* lpszDevice, SDiskCopyOptions cDiskCopyOptions);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
SDiskCopyOptions cDiskCopyOptions	A structure SDiskCopyOptions ( <a href="#">see page 24</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will copy a disc directly to a selected burner.

**Direct access version.**

## [\*\*new\*\*](#) 1.2.32 DACreateImage Function [\*\*new\*\*](#)

**C++**

```
int32 BS_CALL DACreateImage(const TCHAR* lpszDevice, SCreateImageParams cCreateImageParams,
int8 nTaskType);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
SCreateImageParams cCreateImageParams	A structure SCreateImageParams ( <a href="#">see page 14</a> ).
int8 nTaskType	The type of operation. Possible values are: BS_IMGTASK_CREATE ( <a href="#">see page 358</a> ) BS_IMGTASK_VERIFY ( <a href="#">see page 358</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will copy a disc to a image file.

**Direct access version.**

## 1.2.33 DAEjectDevice Function [\*\*new\*\*](#)

**C++**

```
int32 BS_CALL DAEjectDevice(const TCHAR* lpszDevice);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function opens the current device (tray).

**Direct access version.**

## 1.2.34 DAErase Function new

**C++**

```
int32 BS_CALL DAErase(const TCHAR* lpszDevice, BS_BOOL bFast, BS_BOOL bEjectAfterErase);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
BS_BOOL bFast	This parameter states whether the medium is erased using the fast or complete method. Pass BS_TRUE (see page 499) to fast erase the medium, and BS_FALSE (see page 335) to erase the medium completely (this method lasts much longer).
BS_BOOL bEjectAfterErase	This parameter states whether the medium will be ejected after the erase process is complete.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function erases the disc in the current device.

**Direct access version.**

**Notes**

In most cases, it is recommended to fast erase the disc.

Important: On Windows OS it is recommended to eject the medium after erase is complete. Else Windows have problems to recognize that the disc is empty / changed.

## 1.2.35 DAGetBurnSpeed Function new

**C++**

```
int32 BS_CALL DAGetBurnSpeed(const TCHAR* lpszDevice, int32* pnSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pnSpeed	Pointer to the current writing speed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the writing speed of the selected device.

**Direct access version.**

## 1.2.36 DAGetDeviceCapabilities Function new

**C++**

```
int32 BS_CALL DAGetDeviceCapabilities(const TCHAR* lpszDevice, int64* pnDeviceCapabilities);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int64* pnDeviceCapabilities	A device capability to test. See Capabilities.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

If you need to know the device capabilities use this function. **Direct access version.**

**DEPRECATED! IS RELPACED BY:** AnalyseDeviceCapability (see page 120)()

## 1.2.37 DAGetDeviceCapabilitiesHandle Function new

**C++**

```
int32 BS_CALL DAGetDeviceCapabilitiesHandle(const TCHAR* lpszDevice, int32* pnNewCapabilitiesHandle);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pnNewCapabilitiesHandle	An empty handle to the capabilities for the future use.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will provide a capability handle according to the given device index. Call this function before you call AnalyseDeviceCapability (see page 120)().

**Direct access version.**

## 1.2.38 DAGetDeviceInformation Function new

**C++**

```
int32 BS_CALL DAGetDeviceInformation(const TCHAR* lpszDevice, TCHAR chVendorID[9], TCHAR chProductID[17], TCHAR chProductRevision[5]);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
TCHAR chVendorID[9]	Vendor ID of the current device.
TCHAR chProductID[17]	Product ID of the current device.
TCHAR chProductRevision[5]	Product revision of the current device.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the device information of a device from the internal device list.

**Direct access version.**

## 1.2.39 DAGetDeviceInformationEx Function new

**C++**

```
int32 BS_CALL DAGetDeviceInformationEx(const TCHAR* lpszDevice, SExtendedDeviceInformation* pExtendedInformation);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
SExtendedDeviceInformation* pExtendedInformation	A pointer to a SExtendedDeviceInformation (see page 29) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the device information of a device from the internal device list.

**Direct access version.**

## 1.2.40 DAGetMaxBurnSpeed Function new

**C++**

```
int32 BS_CALL DAGetMaxBurnSpeed(const TCHAR* lpszDevice, int32* pnSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pnSpeed	Pointer to the max. writing speed of the current device.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the max. writing speed of the current device.

**Direct access version.**

**Notes**

The IsoSDK returns only the highest value. You may want to use a different speed setting. Valid speed settings are usually calculated this way: 1x, 2x as default, 4 and multiples of 4 till the max. burning speed is reached.

## 1.2.41 DAGetMaxReadSpeed Function new

**C++**

```
int32 BS_CALL DAGetMaxReadSpeed(const TCHAR* lpszDevice, int32* pnSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pnSpeed	Pointer to the max. reading speed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the max. reading speed of the current device.

**Direct access version.**

## 1.2.42 DAGetMediumFreeId Function new

**C++**

```
int32 BS_CALL DAGetMediumFreeId(const TCHAR* lpszDevice, TCHAR id[9]);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
TCHAR id[9]	The id (length 9) of the audio disc.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function will receive the unique disc id for the audio disc.

**Direct access version.**

## 1.2.43 DAGetMediumInformation Function new

**C++**

```
int32 BS_CALL DAGetMediumInformation(const TCHAR* lpszDevice, struct SMediumInfo* pMediumInfo);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
struct SMediumInfo* pMediumInfo	A pointer to a tagMediumInformation structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the information about the current disc in the selected device.

**Direct access version.**

**Notes**

This function returns a valid value only in case of the presence of a disc in the current device. Check first, if there is a disc in

that device.

## 1.2.44 DAGetPossibleBurnSpeeds Function new

### C++

```
int32 BS_CALL DAGetPossibleBurnSpeeds(const TCHAR* lpszDevice, struct SSpeed* pSpeeds,
int32* pnSize);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
struct SSpeed* pSpeeds	A pointer to a SSpeed ( <a href="#">see page 90</a> ) structure.
int32* pnSize	Pointer to the size of the array.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns a structure of the available burning speeds of the current device.

**Direct access version.**

### Notes

This function may need much time. The time is equal to the available speeds, the device has. So DVD speeds are returned very fast, whereas CD speeds need more time.

## 1.2.45 DAGetPossibleImageFormats Function new

### C++

```
int32 BS_CALL DAGetPossibleImageFormats(const TCHAR* lpszDevice, int16* pImageFormats);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int16* plImageFormats	A pointer to the possible image formats.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will check what kind of image formats are possible to save from selected disc.

**Direct access version.**

**Example**

```
int16 nImageFormats;
::GetPossibleImageFormats(&nImageFormats);
if (nImageFormats & BS_IMG_ISO)
{
    //ISO is possible
}
if (nImageFormats & BS_IMG_BIN)
{
    //BIN is possible
}
```

## 1.2.46 DAGetPossibleReadSpeeds Function [new](#)

**C++**

```
int32 BS_CALL DAGetPossibleReadSpeeds(const TCHAR* lpszDevice, struct SSpeed* pSpeeds,
int32* pnSize);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
struct SSpeed* pSpeeds	A pointer to a SSpeed ( <a href="#">see page 90</a> ) structure.
int32* pnSize	Pointer to the size of the array.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns a structure of the available reading speeds of the current device.

**Direct access version.**

## 1.2.47 DAGetReadSpeed Function [new](#)

**C++**

```
int32 BS_CALL DAGetReadSpeed(const TCHAR* lpszDevice, int32* pnSpeed);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pnSpeed	Pointer to the current reading.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the reading speed of the current device.

**Direct access version.**

## 1.2.48 DAGetSessionInformation Function new

**C++**

```
int32 BS_CALL DAGetSessionInformation(const TCHAR* lpszDevice, int32 nSession, struct SSessionInfo* pSessionInfo);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nSession	Session index from the session list you want to check.
struct SSessionInfo* pSessionInfo	A pointer to a SSessionInfo ( <a href="#">see page 88</a> ) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the session information of the current disc in the current device.

**Direct access version.**

**Notes**

This function returns a valid value only in case of the presence of a disc in the current device. You get the session list by using the function GetMediumInformation ([see page 172](#))

## 1.2.49 DAGetTrackFormatEx Function new

**C++**

```
int32 BS_CALL DAGetTrackFormatEx(const TCHAR* lpszDevice, int32 nTrack, int8* pFormat);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nTrack	The track index.

int8* pFormat	Defines the track format type for a RAW project type. Possible values are: BS_RTF_AUDIO (see page 377) BS_RTF_MODE1 (see page 377) BS_RTF_MODE2_FORM1 (see page 377) BS_RTF_MODE2_FORM2 (see page 378) BS_RTF_MODE2_FORMLESS (see page 378)
---------------	---

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the type of the disc track of the current medium of the given device.

**Direct access version.**

## 1.2.50 DAGetTrackIndexes Function new

**C++**

```
int32 BS_CALL DAGetTrackIndexes(const TCHAR* lpszDevice, int32 nTrack, int32* pArrIndexes,
int32* pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nTrack	The index of the track on the disc.
int32* pArrIndexes	A pointer to an int array for the available indexes.
int32* pnLength	The length of the int array.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns a list of sub indexes of the give audio track.

**Direct access version.**

## 1.2.51 DAGetTrackInformation Function new

**C++**

```
int32 BS_CALL DAGetTrackInformation(const TCHAR* lpszDevice, int32 nTrack, struct
STrackInfo* pTrackInfo);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nTrack	Track index from the track list you want to check.
struct STrackInfo* pTrackInfo	A pointer to a STrackInfo (see page 94) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the track information of the current disc in the current device.

**Direct access version.**

## 1.2.52 DAGetTrackISRC Function new

**C++**

```
int32 BS_CALL DAGetTrackISRC(const TCHAR* lpszDevice, int32 nTrack, TCHAR pStrISRC[12]);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nTrack	The index of the track on the disc.
TCHAR pStrISRC[12]	The ISRC (length 12) of the audio track.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the ISRC code from the give audio track.

**Direct access version.**

## 1.2.53 DAGrabAudioTrack Function new

**C++**

```
int32 BS_CALL DAGrabAudioTrack(const TCHAR* lpszDevice, SAudioGrabbingParams cAudioGrabbingParams, int16 nTrackNumber, const TCHAR* strSavePath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
SAudioGrabbingParams cAudioGrabbingParams	A SAudioGrabbingParams (see page 2) structure.

int16 nTrackNumber	The index of the track on the disc.
const TCHAR* strSavePath	The full path where to store the file.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

The function will grab/save a audio track to a file. Tagging and different encoding types are available.

**Direct access version.**

## 1.2.54 DAImportFile Function new

**C++**

```
int32 BS_CALL DAImportFile(const TCHAR* lpszDevice, HSESSION hSession, const TCHAR*
lpszSourcePath, const TCHAR* lpszDestFolderPath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
HSESSION hSession	Handle to an open disc session.
const TCHAR* lpszSourcePath	Pointer to the null-terminated string that names the imported file.
const TCHAR* lpszDestFolderPath	Pointer to the null-terminated string that names the destination file of the project.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function imports a file from a session to your project.

**Direct access version.**

## 1.2.55 DAImportFileEx Function new

**C++**

```
int32 BS_CALL DAImportFileEx(const TCHAR* lpszDevice, HSESSION hSession, const TCHAR*
lpszSourcePath, const TCHAR* lpszDestFolderPath, const TCHAR* pchPassword);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
HSESSION hSession	Handle to an open disc session

const TCHAR* lpszSourcePath	Pointer to the null-terminated string that names the imported file
const TCHAR* lpszDestFolderPath	Pointer to the null-terminated string that names the destination file of the project
const TCHAR* pchPassword	Pointer to the null-terminated string that holds the password to encrypt.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function imports a file from a session to your project.

**Direct access version.**

## 1.2.56 DAIsDeviceReady Function new

**C++**

```
int32 BS_CALL DAIsDeviceReady(const TCHAR* lpszDevice, BS_LP_BOOL pbReady);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
BS_LP_BOOL pbReady	Pointer to a BOOL value that states whether the current device is ready.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function tests if the current device is ready.

**Direct access version.**

## 1.2.57 DALockMedium Function new

**C++**

```
int32 BS_CALL DALockMedium(const TCHAR* lpszDevice, BS_BOOL bLock);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
BS_BOOL bLock	A BOOL value to lock(TRUE) or unlock (FALSE) the disc.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will lock/unlock a disc inside a selected device.

**Direct access version.**

## 1.2.58 DAOPENDISKSESSION Function new

**C++**

```
int32 BS_CALL DAOPENDISKSESSION(const TCHAR* lpszDevice, int32 nTrackNumber, HSESSION* phSession, int32 nFileSystem);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nTrackNumber	Number of the track to be opened.
HSESSION* phSession	Pointer to the returned session handle.
int32 nFileSystem	The file system you want to open. The following values are allowed: BS_FS_UNKNOWN (see page 338) BS_FS_ISO9660 (see page 337) BS_FS_JOLIET (see page 337) BS_FS_UDF (see page 337) BS_FS_BOOTABLE (see page 336)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function opens a session on the selected disc/device.

**Direct access version.**

## 1.2.59 DAPLAYAUDIOTRACK Function new

**C++**

```
int32 BS_CALL DAPLAYAUDIOTRACK(const TCHAR* lpszDevice, int16 nTrackNumber);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

int16 nTrackNumber	The tracknumber (index) you want to play.
--------------------	---

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will play a audio track on the current disc or disc image.

**Direct access version.**

## 1.2.60 DAPrepare Function [new]

**C++**

```
int32 BS_CALL DAPrepare(const TCHAR* lpszDevice);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables.

**Direct access version.**

**Notes**

This function maybe need some time to return if you manage many small files on the disc.

## 1.2.61 DAReadCDText Function [new]

**C++**

```
int32 BS_CALL DAReadCDText(const TCHAR* lpszDevice, int32* pHandle);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32* pHandle	A handle to receive the information.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will create a handle to the CD-Text information of the audio disc according to the give device index.

**Direct access version.**

## 1.2.62 DAReadFileContents Function new

**C++**

```
int32 BS_CALL DAReadFileContents(const TCHAR* lpszDevice, HSESSION hSession, const TCHAR* lpszFilePath, int64 nOffset, void* pBuffer, int32 nBufferSize, int32* pRead);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
HSESSION hSession	Handle of opened session.
const TCHAR* lpszFilePath	File path on disc (examples: "file.bin", "\Dir1\Dir2\file.ext").
int64 nOffset	Offset in file.
void* pBuffer	A pointer to a output buffer.
int32 nBufferSize	Output buffer size in bytes = number of bytes to read.
int32* pRead	Number of bytes readed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads a segment of file from disc to a buffer.

**Direct access version.**

## 1.2.63 DAReadSectors Function new

**C++**

```
int32 BS_CALL DAReadSectors(const TCHAR* lpszDevice, int32 nLBA, int32 nCount, int8 nFormat, void* pBuff, int32 nBuffLength);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nLBA	The current LBA to read.
int32 nCount	Number of tries to read.
int8 nFormat	Format to read. RAW (1) or normal (0).
void* pBuff	A pointer to the buffer you want to read to.

int32 nBuffLength	The length of the readed buffer. Possible are 2048 for default reading or 2352 for raw reading.
-------------------	---

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will read a given sector into a buffer.

**Direct access version.**

## 1.2.64 DASaveTrackToFile Function new

**C++**

```
int32 BS_CALL DASaveTrackToFile(const TCHAR* lpszDevice, int16 nTrackNumber, const TCHAR* pchFileName, int32 nFormat);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int16 nTrackNumber	The number of the track inside the disc or image file.
const TCHAR* pchFileName	The path and name of the target file.
int32 nFormat	The fileformat of the track. Following values are possible: BS_FF_WAVE (see page 336) BS_FF_ISO (see page 336) BS_FF_BIN (see page 335) BS_FF_MPEG (see page 336)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function saves a selected track to a file.

**Direct access version.**

## 1.2.65 DASetBurnSpeed Function new

**C++**

```
int32 BS_CALL DASetBurnSpeed(const TCHAR* lpszDevice, int32 nSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

int32 nSpeed	The new read speed value. If you only want to use the max. reading speed, pass the value BS_MAX_SPEED (see page 364). This is the default value
--------------	---

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the writing speed of the current device.

**Direct access version.**

**Notes**

If you use multiple devices (multi-device burning) this option will set the speed for all devices. If a device will not support the speed, it will be set to the nearest speed to this value.

## 1.2.66 DASetReadSpeed Function new

**C++**

```
int32 BS_CALL DASetReadSpeed(const TCHAR* lpszDevice, int32 nSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
int32 nSpeed	The new read speed value. If you only want to use the max. reading speed, pass the value BS_MAX_SPEED (see page 364). This is the default value

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the reading speed of the current device.

**Direct access version.**

## 1.2.67 DASetRegionalCode Function new

**C++**

```
int32 BS_CALL DASetRegionalCode(const TCHAR* lpszDevice, int32 nRPC);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.

int32 nRPC	The new region code to be set.
------------	--------------------------------

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0.

**Direct access version.**

**Notes**

The change of a region code is limited! So if you allow this function in your software make sure to inform the user about this. You can get the "changes left" with the function GetDeviceInformationEx (see page 163()).

## 1.2.68 DAVerifyFile Function new

**C++**

```
int32 BS_CALL DAVerifyFile(const TCHAR* lpszDevice, HSESSION hSession, const TCHAR* lpszSourcePath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The name of the device to be selected.
HSESSION hSession	The session where you want to receive the information from.
const TCHAR* lpszSourcePath	The path to the file or a path to a folder on the disc.. Example "\data\testfile.txt"

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function verify a file of the medium against whether it is is readable without any error.

**Direct access version.**

## 1.2.69 DeInitALL Function

**C++**

```
void BS_CALL DeInitALL();
```

**File**

IsoSDKExport.h (see page 577)

**Description**

This is function DeInitALL.

## 1.2.70 DeInitialize Function

### C++

```
int32 BS_CALL DeInitialize();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function de-initializes the IsoSDK. Components of the IsoSDK will be released from memory.

---

## 1.2.71 DeleteProject Function

### C++

```
int32 BS_CALL DeleteProject();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function deletes the current project.

### Notes

Use this function only when terminating the program or if you want to create a new project.

---

## 1.2.72 DirExists Function

### C++

```
int32 BS_CALL DirExists(const TCHAR * strDirectoryName);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * strDirectoryName	Directory name to check

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will check whether a directory exists inside the project tree.

---

## 1.2.73 EjectDevice Function

**C++**

```
int32 BS_CALL EjectDevice(int32 index);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function opens the current device (tray).

---

## 1.2.74 EnableImageDevice Function

**C++**

```
int32 BS_CALL EnableImageDevice(BS_BOOL bEnable);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
BS_BOOL bEnable	This value will enable or disable the internal ImageWriter. The ImageWriter is used to create ISO images from your projects.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Enable or disable the internal ImageWriter device inside the device list.

## 1.2.75 EnableMCNDisabling Function

### C++

```
int32 BS_CALL EnableMCNDisabling(BS_BOOL bEnable);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
BS_BOOL bEnable	A BOOL value to disable (FALSE) or enable (TRUE) the media change notification.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will enable / disable the media change notification event on windows driven computers. Use this function to work with robotic systems.

## 1.2.76 Erase Function

### C++

```
int32 BS_CALL Erase(BS_BOOL bFast, BS_BOOL bEjectAfterErase);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
BS_BOOL bFast	This parameter states whether the medium is erased using the fast or complete method. Pass BS_TRUE ( <a href="#">see page 499</a> ) to fast erase the medium, and BS_FALSE ( <a href="#">see page 335</a> ) to erase the medium completely (this method lasts much longer).
BS_BOOL bEjectAfterErase	This parameter states whether the medium will be ejected after the erase process is complete.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function erases the disc in the current device.

### Notes

In most cases, it is recommended to fast erase the disc.

**Important:** On Windows OS it is recommended to eject the medium after erase is complete. Else Windows have problems to recognize that the disc is empty / changed.

## 1.2.77 EraseDialog Function

### C++

```
int32 BS_CALL EraseDialog(const TCHAR * pchTitle);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>const</b> TCHAR * <b>pchTitle</b>	With this value you can display your own window title in the dialog. Pass NULL, if you do not want to use this feature.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function starts the deletion process with the embedded dialog.

## 1.2.78 EraseMpegByIndex Function

### C++

```
int32 BS_CALL EraseMpegByIndex(int32 index);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>int32</b> <b>index</b>	This value represents the position in the file list. It starts from 0 to x.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function deletes a previously added MPEG file from a VCD or SVCD project.

### Notes

First, check with GetMpegCount ([see page 172](#)()) if files are available.

## 1.2.79 ForceDeInitialize Function

### C++

```
int32 BS_CALL ForceDeInitialize(BS_BOOL bForce);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Description**

This is function ForceDeInitialize.

## 1.2.80 GetActiveDevicesCount Function

**C++**

```
int32 BS_CALL GetActiveDevicesCount(int32 * pnCount);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 * pnCount	A valid pointer to a variable to save the device count to.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.

---

## 1.2.81 GetASPI Function

**C++**

```
int32 BS_CALL GetASPI(int32* pAspiInterface);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32* pAspiInterface	A pointer to an int value. You will get the currently set ASPI layer: BS_ASPI_INTERNAL ( <a href="#">see page 310</a> ) BS_ASPI_WNASPI ( <a href="#">see page 310</a> ) BS_ASPI_FROGASPI ( <a href="#">see page 309</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

With this function you can get the information about the currently used ASPI layer.

## 1.2.82 GetAudioFileSize Function

### C++

```
int32 BS_CALL GetAudioFileSize(const TCHAR * strFileName, double * fSize);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * strFileName	The file to calculate the file size. File name with full path.
double * fSize	Pointer to the file size value.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function enumerate the final size of the compressed audio file

### Notes

Use this function only with Ogg, Mp3, WMA, Mp4, Flac and Wav files. Other files will cause an exception.

Make sure that you stop playing a file before you de-init the SDK. Otherwise you will get an exception.

---

## 1.2.83 GetAudioTags Function

### C++

```
int32 BS_CALL GetAudioTags(const TCHAR* strFileName, const char* tags);
```

### File

IsoSDKExport.h ([see page 577](#))

### Description

This is function GetAudioTags.

---

## 1.2.84 GetBootInfoEx Function

### C++

```
int32 BS_CALL GetBootInfoEx(struct SBootInfoEx * pBootInfoEx);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SBootInfoEx * pBootInfoEx	A structure SBootInfoEx ( <a href="#">see page 6</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.

**Notes**

All settings are part of the “El Torito” specification. Please make sure that you are familiar with them in order to use the settings.

These settings will be reset after a new call of a data project.

## 1.2.85 GetBootVolumeInformation Function

**C++**

```
int32 BS_CALL GetBootVolumeInformation(HSESSION hSession, struct SBootVolumeInfo* pBootVolumeInfo);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	The session where you want to receive the information from.
struct SBootVolumeInfo* pBootVolumeInfo	A pointer to a SBootVolumeInfo (see page 10) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to receive information about the boot file system on a existing disc or image.

## 1.2.86 GetBurnDevice Function

**C++**

```
int32 BS_CALL GetBurnDevice(TCHAR* lpszDevice, int32* pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR* lpszDevice	Pointer to the null-terminated string with the device description.
int32* pnLength	Pointer to the length of the string.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the currently selected burn device. If you use multiple devices this function will return more than one device. Check the array with GetActiveDeviceCount()

**Notes**

Before you can use this function, you must have called SetDevice() first.

## 1.2.87 GetBurnDevices Function

**C++**

```
int32 BS_CALL GetBurnDevices(TCHAR chListDevices[26][50], int32 * pnUsed);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
TCHAR chListDevices[26][50]	Array with the current selected burn devices.
int32 * pnUsed	Number of devices inside the array.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns an array with all selected burn devices.

**Example**

```
TCHAR chBurnDevices[26][50];
int32 nBurnDevices = 0;
int32 res = ::GetBurnDevices(chBurnDevices, &nBurnDevices);
```

## 1.2.88 GetBurnDoneEventCallback Function

**C++**

```
BurnDoneEvent BS_CALL GetBurnDoneEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original BurnDoneEvent ( <a href="#">see page 237</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set BurnDoneEvent ([see page 237](#)). Use this if you need the event in another dialog.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

## 1.2.89 GetBurnFileEventCallback Function

**C++**

```
BurnFileEvent BS_CALL GetBurnFileEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original BurnFileEvent ( <a href="#">see page 237</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set BurnFileEvent ([see page 237](#)). Use this if you need the event in another dialog.

## 1.2.90 GetBurnSpeed Function

**C++**

```
int32 BS_CALL GetBurnSpeed(int32 index, int32 * pnSpeed);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int32 * pnSpeed	Pointer to the current writing speed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the writing speed of the selected device.

## 1.2.91 GetCDTextDiskTagString Function

### C++

```
int32 BS_CALL GetCDTextDiskTagString(int32 hOpenedHandle, int32 nCDTCI, TCHAR* pchText,
int32 * pnLength);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 hOpenedHandle	The handle that was created with ReadCDText ( <a href="#">see page 194</a> ).
int32 nCDTCI	The field to receive. Possible values are: BS_CDTCL_ARRANGER ( <a href="#">see page 319</a> ) BS_CDTCL_COMPOSER ( <a href="#">see page 319</a> ) BS_CDTCL_MESSAGE ( <a href="#">see page 319</a> ) BS_CDTCL_PERFORMER ( <a href="#">see page 320</a> ) BS_CDTCL_SONG_WRITER ( <a href="#">see page 320</a> ) BS_CDTCL_TITLE ( <a href="#">see page 320</a> )
TCHAR* pchText	A buffer to the CD-Text information to receive.
int32 * pnLength	The buffer length of pchItemText.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will get a specified CD-Text item from the created CD-Text handle of a audio disc.

### Notes

pchItemText first should be passed equal to 0, to learn the required buffer length, stored in pnLen.

## 1.2.92 GetCDTextTrackTagString Function

### C++

```
int32 BS_CALL GetCDTextTrackTagString(int32 hOpenedHandle, int32 nTrackNumber, int32
nCDTCI, TCHAR* pchText, int32 * pnLength);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 hOpenedHandle	The handle that was created with ReadCDText ( <a href="#">see page 194</a> ).
int32 nTrackNumber	The index of the Track on the disc.

int32 nCDTCI	The field to receive. Possible values are: BS_CDTCI_ARRANGER (see page 319) BS_CDTCI_COMPOSER (see page 319) BS_CDTCI_MESSAGE (see page 319) BS_CDTCI_PERFORMER (see page 320) BS_CDTCI_SONG_WRITER (see page 320) BS_CDTCI_TITLE (see page 320)
TCHAR* pchText	A buffer to the CD-Text information to receive.
int32 * pnLength	The buffer length of pchItemText.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will get a specified CD-Text track item from the created CD-Text handle of a audio disc.

**Notes**

pchItemText first should be passed equal to 0, to learn the required buffer length, stored in pnLen.

## 1.2.93 GetCompareFilesForArrangementEventCallback Function

**C++**

```
CompareFilesForArrangementEvent BS_CALL GetCompareFilesForArrangementEventCallback(void** pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void** pUserData	A pointer to the user data of the original CompareFilesForArrangementEvent (see page 237).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set CompareFilesForArrangementEvent (see page 237). Use this if you need the event in another dialog.

## 1.2.94 GetCompressEncrypt Function

**C++**

```
int32 BS_CALL GetCompressEncrypt(struct SCompressEncryptInfo* pCEInfo);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SCompressEncryptInfo* pCEInfo	A pointer to the SCompressEncryptInfo (see page 11) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will get the parameters for internal compression and encryption.

## 1.2.95 GetDeviceCapabilities Function

**C++**

```
int32 BS_CALL GetDeviceCapabilities(int32 index, int64 * pnDeviceCapabilities);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int64 * pnDeviceCapabilities	A device capability to test. See Capabilities.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

If you need to know the device capabilities use this function.

**DEPRECATED! IS REPLACED BY:** AnalyseDeviceCapability (see page 120)()

## 1.2.96 GetDeviceCapabilitiesHandle Function

**C++**

```
int32 BS_CALL GetDeviceCapabilitiesHandle(int32 index, int32 * pnNewCapabilitiesHandle);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 * pnNewCapabilitiesHandle	An empty handle to the capabilities for the future use.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will provide a capability handle according to the given device index. Call this function before you call AnalyseDeviceCapability (see page 120)().

## 1.2.97 GetDeviceInformation Function

**C++**

```
int32 BS_CALL GetDeviceInformation(int32 index, TCHAR chVendorID[9], TCHAR chProductID[17],
TCHAR chProductRevision[5]);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
TCHAR chVendorID[9]	Vendor ID of the current device.
TCHAR chProductID[17]	Product ID of the current device.
TCHAR chProductRevision[5]	Product revision of the current device.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the device information of a device from the internal device list.

## 1.2.98 GetDeviceInformationEx Function

**C++**

```
int32 BS_CALL GetDeviceInformationEx(int32 index, SExtendedDeviceInformation*
pExtendedInformation);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
SExtendedDeviceInformation* pExtendedInformation	A pointer to a SExtendedDeviceInformation (see page 29) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the device information of a device from the internal device list.

## 1.2.99 GetDevices Function

**C++**

```
int32 BS_CALL GetDevices(TCHAR chListDevices[26][50], int32 * pnUsed, BS_BOOL bBurningDevicesOnly);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR chListDevices[26][50]	Multi-dimensional string array with the device descriptions.
int32 * pnUsed	Number of found devices in the array chListDevices.
BS_BOOL bBurningDevicesOnly	Pass BS_TRUE (see page 499) to list only devices with burning capabilities. A BS_FALSE (see page 335) lists all compatible devices regardless of their burning capabilities.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns a list of all available devices or a list of available devices with burning capabilities.

## 1.2.100 GetDVDVideoOptions Function

**C++**

```
int32 BS_CALL GetDVDVideoOptions(struct SDVDVideoOptions* pDVDVideoOptions);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SDVDVideoOptions* pDVDVideoOptions	A pointer to a SDVDVideoOptions (see page 28) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function gets the special options for Video DVD projects.

## 1.2.101 GetEraseDoneEventCallback Function

### C++

```
EraseDoneEvent BS_CALL GetEraseDoneEventCallback(void ** pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>void</b> ** <b>pUserData</b>	A pointer to the user data of the original EraseDoneEvent ( <a href="#">see page 238</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes)

### Description

Hooks an already set EraseDoneEvent ([see page 238](#)). Use this if you need the event in another dialog

---

## 1.2.102 GetErrorDeviceName Function

### C++

```
int32 BS_CALL GetErrorDeviceName(TCHAR* pDeviceName, int32 nLength);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>TCHAR</b> * <b>pDeviceName</b>	Name of the device that reported the last error.
<b>int32</b> <b>nLength</b>	Length of the string.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will return the name of the drive that reported the last error. This is helpful if you use multi-device burning.

---

## 1.2.103 GetFileAllocationTable Function

### C++

```
int32 BS_CALL GetFileAllocationTable(HSESSION hSession, const TCHAR* lpszFilePath, struct  
SFileAllocationTable * pAllocTable, int32 * pnTableSize);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
HSESSION hSession	Handle to an open session.
const TCHAR* lpszFilePath	File name + full path of the file to get the file allocation table from.
struct SFileAllocationTable * pAllocTable	A pointer to a structure SFileAllocationTable (see page 232).
int32 * pnTableSize	The size of the SFileAllocationTable (see page 232) in memory.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to receive information of the file allocation table and the file extents on the disc according to the given file.

## 1.2.104 GetFileEntry Function

**C++**

```
int32 BS_CALL GetFileEntry(const TCHAR * lpszFilePath, struct SFileEntry* pFileEntry);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR * lpszFilePath	File name + full path of the file to get the information from.
struct SFileEntry* pFileEntry	A pointer to a structure SFileEntry (see page 40).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will allow you to get all information about an already added file.

## 1.2.105 GetFileTimeEx Function new

**C++**

```
int32 BS_CALL GetFileTimeEx(struct SFileTimeEx* FileTimeEx);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SFileTimeEx* FileTimeEx	A pointer to the SFileTimeEx (see page 44) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to read the extended FileTimeEx data for the project types.

## 1.2.106 GetFinalizeEventCallback Function

**C++**

```
FinalizeEvent BS_CALL GetFinalizeEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original FinalizeEvent (see page 239).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set FinalizeEvent (see page 239). Use this if you need the event in another dialog.

---

## 1.2.107 GetImagePath Function

**C++**

```
int32 BS_CALL GetImagePath(TCHAR* pchPath, int32* pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR* pchPath	A buffer to the path information to receive.
int32* pnLength	The buffer length of pchPath.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will received the path to a opened image file.

## 1.2.108 GetImageSize Function

### C++

```
int32 BS_CALL GetImageSize(double * pdImageSize);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
double * pdImageSize	This function returns the project size in bytes. The return parameter is a Double value.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns the project size. The GetImageSize function is only available after the execution of the Prepare ([see page 194](#)) function.

### Notes

The functions Prepare ([see page 194](#))() must have been called before this function call.

## 1.2.109 GetInfoTextEventCallback Function

### C++

```
InfoTextEvent BS_CALL GetInfoTextEventCallback(void ** pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
void ** pUserData	A pointer to the user data of the original InfoTextEvent ( <a href="#">see page 240</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Hooks an already set InfoTextEvent ([see page 240](#)). Use this if you need the event in another dialog.

## 1.2.110 GetISOInfoEx Function

### C++

```
int32 BS_CALL GetISOInfoEx(struct SISOInfoEx * pISOInfoEx);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
struct SISOInfoEx * pSISOInfoEx	A pointer to the SISOInfoEx ( <a href="#">see page 56</a> ) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to read the extended ISO fields for Data CD/DVD/BD and ISO images. Available only in data projects.

**Notes**

Very Important: Since Joliet is 2 Bytes, the max. usable length of the strings in this structure is only the reservation length divided by 2. Otherwise your strings will be cut.

## 1.2.111 GetISOVolumeInformation Function

**C++**

```
int32 BS_CALL GetISOVolumeInformation(HSESSION hSession, struct SISOVolumeInfo*
pISOVolumeInfo);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
HSESSION hSession	The session where you want to receive the information from.
struct SISOVolumeInfo* pISOVolumeInfo	A pointer to a SISOVolumeInfo ( <a href="#">see page 64</a> ) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to receive information about the ISO filesystem on a existing disc or image.

## 1.2.112 GetJobDoneEventCallback Function

**C++**

```
JobDoneEvent BS_CALL GetJobDoneEventCallback(void ** pUserData);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original JobDoneEvent ( <a href="#">see page 241</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set JobDoneEvent (see page 241). Use this if you need the event in another dialog.

## 1.2.113 GetLanguage Function

**C++**

```
int32 BS_CALL GetLanguage(TCHAR chLanguage[100], TCHAR chLanguageFile[_MAX_PATH]);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR chLanguage[100]	States the currently used language section of the INI file.
TCHAR chLanguageFile[_MAX_PATH]	File name of language INI file.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to determine the current language of the IsoSDK.

**Notes**

Set the language of the IsoSDK as early as possible.

## 1.2.114 GetMaxBurnSpeed Function

**C++**

```
int32 BS_CALL GetMaxBurnSpeed(int32 index, int32 * pnSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 * pnSpeed	Pointer to the max. writing speed of the current device.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the max. writing speed of the current device.

**Notes**

The IsoSDK returns only the highest value. You may want to use a different speed setting. Valid speed settings are usually calculated this way: 1x, 2x as default, 4 and multiples of 4 till the max. burning speed is reached.

## 1.2.115 GetMaxReadSpeed Function

**C++**

```
int32 BS_CALL GetMaxReadSpeed(int32 index, int32 * pnSpeed);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int32 * pnSpeed	Pointer to the max. reading speed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads the max. reading speed of the current device.

---

## 1.2.116 GetMediumFreedbId Function

**C++**

```
int32 BS_CALL GetMediumFreedbId(int32 index, TCHAR id[9]);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
TCHAR id[9]	The id (length 9) of the audio disc.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function will receive the unique disc id for the audio disc.

## 1.2.117 GetMediumInformation Function

### C++

```
int32 BS_CALL GetMediumInformation(int32 index, struct SMediumInfo * pMediumInfo);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
struct SMediumInfo * pMediumInfo	A pointer to a tagMediumInformation structure.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns the information about the current disc in the selected device.

### Notes

This function returns a valid value only in case of the presence of a disc in the current device. Check first, if there is a disc in that device.

## 1.2.118 GetMpegCount Function

### C++

```
int32 BS_CALL GetMpegCount();
```

### File

IsoSDKExport.h ([see page 577](#))

### Returns

Returns an integer that will represent the amount of MPEG files.

### Description

This function returns the number of added files from a VCD or SVCD project.

## 1.2.119 GetNetworkDiskTagInt Function

### C++

```
int32 BS_CALL GetNetworkDiskTagInt(int32 hOpenedHandle, int32 nTagsContentId, int32* pNum);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 hOpenedHandle	The handle that was created with TagsFromNetworkDialog (see page 229).
int32 nTagsContentId	The ID of the numeric value to received. BS_TCI_DISK_LENGTH (see page 495) BS_TCI_FRAME_OFFSET (see page 495) BS_TCI_REVISION (see page 496) BS_TCI_YEAR (see page 497)
int32* pNum	A pointer to the numeric value to received.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will received a numeric value of a CDDB disc tag.

## 1.2.120 GetNetworkDiskTagString Function

**C++**

```
int32 BS_CALL GetNetworkDiskTagString(int32 hOpenedHandle, int32 nTagsContentId, TCHAR* pchText, int32 * pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 hOpenedHandle	The Handle that was created with TagsFromNetworkDialog (see page 229).
int32 nTagsContentId	The ID of the value to received. BS_TCI_ARTIST (see page 494) BS_TCI_CATEGORY (see page 494) BS_TCI_DISKID (see page 495) BS_TCI_EXTENDED_INFO (see page 495) BS_TCI_GENRE (see page 496) BS_TCI_SUBMITTED_VIA (see page 496) BS_TCI_TITLE (see page 496)
TCHAR* pchText	A buffer to the Tag/CD-Text information to receive.
int32 * pnLength	The buffer length of pchText.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will received a string value of a CDDB disc tag.

## 1.2.121 GetNetworkTrackTagInt Function

### C++

```
int32 BS_CALL GetNetworkTrackTagInt(int32 hOpenedHandle, int32 nTrackNumber, int32
nTagsContentId, int32* pNum);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 hOpenedHandle	The handle that was created with TagsFromNetworkDialog ( <a href="#">see page 229</a> ).
int32 nTrackNumber	Number of the track to receive data from.
int32 nTagsContentId	The ID of the numeric value to received. BS_TCI_FRAME_OFFSET ( <a href="#">see page 495</a> ) BS_TCI_REVISION ( <a href="#">see page 496</a> ) BS_TCI_YEAR ( <a href="#">see page 497</a> )
int32* pNum	A pointer to the numeric value to received.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will received a numeric value of a CDDB track tag according to the given track.

## 1.2.122 GetNetworkTrackTagString Function

### C++

```
int32 BS_CALL GetNetworkTrackTagString(int32 hOpenedHandle, int32 nTrackNumber, int32
nTagsContentId, TCHAR* pchText, int32 * pnLength);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 hOpenedHandle	The handle that was created with TagsFromNetworkDialog ( <a href="#">see page 229</a> ).
int32 nTrackNumber	Number of the track to receive data from.
int32 nTagsContentId	The ID of the value to received BS_TCI_ARTIST ( <a href="#">see page 494</a> ) BS_TCI_CATEGORY ( <a href="#">see page 494</a> ) BS_TCI_DISKID ( <a href="#">see page 495</a> ) BS_TCI_EXTENDED_INFO ( <a href="#">see page 495</a> ) BS_TCI_GENRE ( <a href="#">see page 496</a> ) BS_TCI_SUBMITTED_VIA ( <a href="#">see page 496</a> ) BS_TCI_TITLE ( <a href="#">see page 496</a> )
TCHAR* pchText	A buffer to the Tag/CD-Text information to receive.

int32 * pnLength	The buffer length of pchText.
------------------	-------------------------------

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will receive a string value of a CDDB track tag according to the given track.

**Notes**

pchTextText first should be passed equal to 0, to learn the required buffer length, stored in pnLength.

## 1.2.123 GetNumberOfFiles Function

**C++**

```
int32 BS_CALL GetNumberOfFiles(HDIR hDirectory, int32* pNumber);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HDIR hDirectory	Handle to an open directory.
int32* pNumber	Pointer to an int value that will receive the number of files.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the number of files of the current disc in the current device.

## 1.2.124 GetOptions Function

**C++**

```
int32 BS_CALL GetOptions(struct SOptions * pOptions);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SOptions * pOptions	A pointer to the SOptions (see page 75) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to read the burning options.

## 1.2.125 GetPlayTime Function

### C++

```
int32 BS_CALL GetPlayTime(const TCHAR * strFileName, int32 * time);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * strFileName	Full file name, with path, of the audio file.
int32 * time	Pointer to the received playtime in seconds.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns the playtime of the given audio file in seconds.

### Notes

Use this function only with Ogg, Mp3, WMA, Mp4, Flac and Wav files. Other files will cause an exception.

## 1.2.126 GetPossibleBurnSpeeds Function

### C++

```
int32 BS_CALL GetPossibleBurnSpeeds(int32 index, struct SSpeed * pSpeeds, int32 * pnSize);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
struct SSpeed * pSpeeds	A pointer to a SSpeed ( <a href="#">see page 90</a> ) structure.
int32 * pnSize	Pointer to the size of the array.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns a structure of the available burning speeds of the current device.

### Notes

This function may need much time. The time is equal to the available speeds, the device has. So DVD speeds are returned very fast, whereas CD speeds need more time.

## 1.2.127 GetPossibleImageFormats Function

### C++

```
int32 BS_CALL GetPossibleImageFormats(int16 * pImageFormats);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int16 * plImageFormats	A pointer to the possible image formats.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will check what kind of image formats are possible to save from selected disc.

### Example

```
int16 nImageFormats;
::GetPossibleImageFormats(&nImageFormats);
if (nImageFormats & BS_IMG_ISO)
{
    //ISO is possible
}
if (nImageFormats & BS_IMG_BIN)
{
    //BIN is possible
}
```

## 1.2.128 GetPossibleReadSpeeds Function

### C++

```
int32 BS_CALL GetPossibleReadSpeeds(int32 index, struct SSpeed * pSpeeds, int32 * pnSize);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
struct SSpeed * pSpeeds	A pointer to a SSpeed ( <a href="#">see page 90</a> ) structure.
int32 * pnSize	Pointer to the size of the array.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns a structure of the available reading speeds of the current device.

## 1.2.129 GetPrecisePlayTime Function

### C++

```
int32 BS_CALL GetPrecisePlayTime(const TCHAR * strFileName, double * time);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * strFileName	Full file name, with path, of the audio file.
double * time	Pointer to the received playtime in milliseconds.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function returns the playtime of the given audio file in milliseconds.

### Notes

Use this function only with Ogg, Mp3, WMA, Mp4, Flac and Wav files. Other files will cause an exception.

## 1.2.130 GetProcessEventCallback Function

### C++

```
ProcessEvent BS_CALL GetProcessEventCallback(void ** pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
void ** pUserData	A pointer to the user data of the original ProcessEvent ( <a href="#">see page 242</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Hooks an already set ProcessEvent ([see page 242](#)). Use this if you need the event in another dialog.

## 1.2.131 GetProjectType Function

### C++

```
int32 BS_CALL GetProjectType(int32 * pnType);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
int32 * pnType	A valid pointer to the project type. The following values are supported: BS_BURNER_AUDIO ( <a href="#">see page 314</a> ) BS_BURNER_CUE ( <a href="#">see page 314</a> ) BS_BURNER_DATA ( <a href="#">see page 315</a> ) BS_BURNER_VCD ( <a href="#">see page 316</a> ) BS_BURNER_SVCD ( <a href="#">see page 316</a> ) BS_BURNER_VIDEODVD ( <a href="#">see page 316</a> ) BS_BURNER_UDFDVD ( <a href="#">see page 316</a> ) BS_BURNER_ISOUDF ( <a href="#">see page 315</a> ) BS_BURNER_BLURAY ( <a href="#">see page 314</a> ) BS_BURNER_MIXEDMODE ( <a href="#">see page 315</a> ) BS_BURNER_RAW ( <a href="#">see page 315</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the type of the current project.

## 1.2.132 GetRAWDataEventCallback Function

**C++**

```
RAWDataEvent BS_CALL GetRAWDataEventCallback(void** pUserData);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
void** pUserData	A pointer to the user data of the original RAWDataEvent ( <a href="#">see page 242</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set RAWDataEvent ([see page 242](#)). Use this if you need the event in another dialog.

## 1.2.133 GetReadDevice Function

**C++**

```
int32 BS_CALL GetReadDevice(TCHAR* lpszDevice, int32* pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR* lpszDevice	Pointer to the null-terminated string with the device description.
int32* pnLength	Later

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the currently selected read device.

**Notes**

Before you can use this function, you must have called SetReadDevice (see page 219)() first.

## 1.2.134 GetReadSpeed Function

**C++**

```
int32 BS_CALL GetReadSpeed(int32 index, int32 * pnSpeed);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 * pnSpeed	Pointer to the current reading.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the reading speed of the current device.

## 1.2.135 GetSessionInformation Function

**C++**

```
int32 BS_CALL GetSessionInformation(int32 index, int32 nSession, struct SSessionInfo * pSessionInfo);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nSession	Session index from the session list you want to check.
struct SSessionInfo * pSessionInfo	A pointer to a SSessionInfo (see page 88) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the session information of the current disc in the current device.

**Notes**

This function returns a valid value only in case of the presence of a disc in the current device. You get the session list by using the function GetMediumInformation (see page 172)

## 1.2.136 GetStartVerifyEventCallback Function

**C++**

```
StartVerifyEvent BS_CALL GetStartVerifyEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original StartVerifyEvent (see page 243).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set StartVerifyEvent (see page 243). Use this if you need the event in another dialog.

## 1.2.137 GetText Function

**C++**

```
int32 BS_CALL GetText(int32 nTextID, TCHAR * pchText, int32 * pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 nTextID	Text ID of the text you want to get.

TCHAR * pchText	Pointer to the text of the provided text ID.
int32 * pnLength	Text length of the text of the pointer pchText.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the text of a text ID according to your language setting; e.g. error code.

## 1.2.138 GetTmpPath Function

**C++**

```
int32 BS_CALL GetTmpPath(TCHAR chPath[_MAX_PATH]);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
TCHAR chPath[_MAX_PATH]	Returns a string that contains the current path.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

To determine the current temporary path, call this function.

## 1.2.139 GetTrackFormatEx Function

**C++**

```
int32 BS_CALL GetTrackFormatEx(int32 index, int32 nTrack, int8 * pFormat);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nTrack	The track index.
int8 * pFormat	Defines the track format type for a RAW project type. Possible values are: BS_RTF_AUDIO (see page 377) BS_RTF_MODE1 (see page 377) BS_RTF_MODE2_FORM1 (see page 377) BS_RTF_MODE2_FORM2 (see page 378) BS_RTF_MODE2_FORMLESS (see page 378)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns the type of the disc track of the current medium of the given device.

## 1.2.140 GetTrackIndexes Function

**C++**

```
int32 BS_CALL GetTrackIndexes(int32 index, int32 nTrack, int32 * pArrIndexes, int32 * pnLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nTrack	The index of the track on the disc.
int32 * pArrIndexes	A pointer to an int array for the available indexes.
int32 * pnLength	The length of the int array.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns a list of sub indexes of the give audio track.

## 1.2.141 GetTrackInformation Function

**C++**

```
int32 BS_CALL GetTrackInformation(int32 index, int32 nTrack, struct STrackInfo * pTrackInfo);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nTrack	Track index from the track list you want to check.
struct STrackInfo * pTrackInfo	A pointer to a STrackInfo (see page 94) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code

will be returned (see Error Codes).

#### Description

This function returns the track information of the current disc in the current device.

## 1.2.142 GetTrackISRC Function

#### C++

```
int32 BS_CALL GetTrackISRC(int32 index, int32 nTrack, TCHAR pStrISRC[12]);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nTrack	The index of the track on the disc.
TCHAR pStrISRC[12]	The ISRC (length 12) of the audio track.

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

This function returns the ISRC code from the give audio track.

## 1.2.143 GetUDFOptions Function

#### C++

```
int32 BS_CALL GetUDFOptions(struct SUDFOptions * pUdfOptions);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
struct SUDFOptions * pUdfOptions	A structure SUDFOptions (see page 97).

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

Use this function to set the UDF options for UDF data projects.

## 1.2.144 GetUDFOptionsEx Function

### C++

```
int32 BS_CALL GetUDFOptionsEx(struct SUDFOptionsEx * pUdfOptions);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SUDFOptionsEx * pUdfOptions	A pointer to the SUDFOptionsEx ( <a href="#">see page 232</a> ) structure.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function to read the UDF extended options for UDF data projects.

## 1.2.145 GetUDFVolumeInformation Function

### C++

```
int32 BS_CALL GetUDFVolumeInformation(HSESSION hSession, struct SUDFVolumeInfo* pUDFVolumeInfo);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
HSESSION hSession	The session where you want to receive the information from.
struct SUDFVolumeInfo* pUDFVolumeInfo	A pointer to a SUDFVolumeInfo ( <a href="#">see page 100</a> ) structure.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function to receive information about the UDF file system on a existing disc or image.

## 1.2.146 GetVerify Function

### C++

```
int32 BS_CALL GetVerify(BS_BOOL * bVerify);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
BS_BOOL * bVerify	States whether the afterburn verification is enabled.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to get the afterburn verification state.

## 1.2.147 GetVerifyDoneEventCallback Function

**C++**

```
VerifyDoneEvent BS_CALL GetVerifyDoneEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original VerifyDoneEvent (see page 244).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.

## 1.2.148 GetVerifyErrorEventCallback Function

**C++**

```
VerifyErrorEvent BS_CALL GetVerifyErrorEventCallback(void ** pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void ** pUserData	A pointer to the user data of the original VerifyDoneEvent (see page 244).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.

## 1.2.149 GetVerifyFileEventCallback Function

### C++

```
VerifyFileEvent BS_CALL GetVerifyFileEventCallback(void ** pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<code>void ** pUserData</code>	A pointer to the user data of the original VerifyFileEvent ( <a href="#">see page 245</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Hooks an already set VerifyFileEvent ([see page 245](#)). Use this if you need the event in another dialog.

## 1.2.150 GetVerifySectorEventCallback Function

### C++

```
VerifySectorEvent BS_CALL GetVerifySectorEventCallback(void ** pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<code>void ** pUserData</code>	A pointer to the user data of the original VerifySectorEvent ( <a href="#">see page 245</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Hooks an already set VerifySectorEvent ([see page 245](#)). Use this if you need the event in another dialog.

## 1.2.151 GetWriteCDTextInUnicode Function

### C++

```
int32 BS_CALL GetWriteCDTextInUnicode(BS_BOOL* pbWriteInUnicode);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
BS_BOOL* pbWriteInUnicode	A pointer to a BOOLEAN value.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to read the CD-Text Unicode switch.

**Notes**

CD-Text itself is defined without Unicode. Some software products (players e.g) support Unicode CD-Text. But most players, plugins and whatever did not support Unicode.

## 1.2.152 GrabAudioTrack Function

**C++**

```
int32 BS_CALL GrabAudioTrack(SAudioGrabbingParams cAudioGrabbingParams, int16 nTrackNumber,
                           const TCHAR * strSavePath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
SAudioGrabbingParams cAudioGrabbingParams	A SAudioGrabbingParams (see page 2) structure.
int16 nTrackNumber	The index of the track on the disc.
const TCHAR * strSavePath	The full path where to store the file.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

The function will grab/save a audio track to a file. Tagging and different encoding types are available.

## 1.2.153 ImportFile Function

**C++**

```
int32 BS_CALL ImportFile(HSESSION hSession, const TCHAR* lpszSourcePath, const TCHAR*
                         lpszDestFolderPath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	Handle to an open disc session.

const TCHAR* lpszSourcePath	Pointer to the null-terminated string that names the imported file.
const TCHAR* lpszDestFolderPath	Pointer to the null-terminated string that names the destination file of the project.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function imports a file from a session to your project.

## 1.2.154 ImportFileEx Function

**C++**

```
int32 BS_CALL ImportFileEx(HSESSION hSession, const TCHAR* lpszSourcePath, const TCHAR * lpszDestFolderPath, const TCHAR* pchPassword);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	Handle to an open disc session.
const TCHAR* lpszSourcePath	Pointer to the null-terminated string that names the imported file.
const TCHAR * lpszDestFolderPath	Pointer to the null-terminated string that names the destination file of the project.
const TCHAR* pchPassword	Pointer to the null-terminated string that holds the password to encrypt.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function imports a file from a session to your project.

## 1.2.155 Initialize Function

**C++**

```
int32 BS_CALL Initialize(const TCHAR chLicenseKey[35], int32 nAspiInterface, BS_BOOL bEnableImageDevice);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR chLicenseKey[35]	The license key you got after purchased the IsoSDK.

int32 nAspiInterface	With this parameter you can chose the ASPI interface you want to use. The IsoSDK supports: BS_ASPI_INTERNAL (see page 310) BS_ASPI_WNASPI (see page 310) BS_ASPI_FROGASPI (see page 309)
BS_BOOL bEnableImageDevice	This parameter will enable/disable the ImageWriter in the drive (device) list. The ImageWriter is the internal device to create ISO images from your projects.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function initializes the IsoSDK. The function must be called before you can use other functions.

**Notes**

The function must be called before you can use other functions.

You can only use the internal ASPI layer with Windows 2000 or higher.

---

## 1.2.156 IsDeviceReady Function

**C++**

```
int32 BS_CALL IsDeviceReady(int32 index, BS_LP_BOOL pbReady);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
BS_LP_BOOL pbReady	Pointer to a BOOL value that states whether the current device is ready.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function tests if the current device is ready.

**Notes**

Before you can use this function, a device must be set first.

---

## 1.2.157 IsValidVideoTsFolder Function

**C++**

```
BS_BOOL BS_CALL IsValidVideoTsFolder(const TCHAR * strPath);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
const TCHAR * strPath	This value contains the full path to the folder where the VideoDVD image is stored. You can include or exclude the VideoTS folder.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will check if a given path contains a valid VideoDVD structure with IFO, BUP and VOB Files.

**Notes**

If you use the IsoSDK inside a software that will burn copied or shrinked Movie DVDs, we suggest to do not use this function. Many current MovieDVD releases contain a corrupted structure.

## 1.2.158 LoadBassPlugin Function

**C++**

```
int32 BS_CALL LoadBassPlugin(const TCHAR * strFileName);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
const TCHAR * strFileName	The file to load. File name with full path.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will load a given Bass plugin to the IsoSDK. Bass.dll have to be present.

<http://www.un4seen.com/>

**NOTE:** The bass.dll and plugins are available as 32 and 64 bit.

## 1.2.159 LockMedium Function

**C++**

```
int32 BS_CALL LockMedium(int32 index, BS_BOOL bLock);
```

**File**IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
BS_BOOL bLock	A BOOL value to lock(TRUE) or unlock (FALSE) the disc.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will lock/unlock a disc inside a selected device.

## 1.2.160 MultiDeviceDialog Function

**C++**

```
int32 BS_CALL MultiDeviceDialog();
```

**File**

IsoSDKExport.h (see page 577)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function starts the multi-device selection dialog to burn on more than one device.

## 1.2.161 OpenDirectory Function

**C++**

```
int32 BS_CALL OpenDirectory(HSESSION hSession, const TCHAR * lpszDirName, HDIR * phDir);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	Handle to an open session.
const TCHAR * lpszDirName	Pointer to the null-terminated string that names the directory to be opened.
HDIR * phDir	Pointer to the returned directory handle.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function opens a directory on the selected session.

## 1.2.162 OpenDiskSession Function

### C++

```
int32 BS_CALL OpenDiskSession(int32 index, int32 nTrackNumber, HSESSION * phSession, int32 nFileSystem);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int32 nTrackNumber	Number of the track to be opened.
HSESSION * phSession	Pointer to the returned session handle.
int32 nFileSystem	The file system you want to open. The following values are allowed: BS_FS_UNKNOWN ( <a href="#">see page 338</a> ) BS_FS_ISO9660 ( <a href="#">see page 337</a> ) BS_FS_JOLIET ( <a href="#">see page 337</a> ) BS_FS_UDF ( <a href="#">see page 337</a> ) BS_FS_BOOTABLE ( <a href="#">see page 336</a> )

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function opens a session on the selected disc/device.

## 1.2.163 PlayAudioFile Function

### C++

```
int32 BS_CALL PlayAudioFile(const TCHAR * strFileName);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * strFileName	The file to play. File name with full path.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function to play the given file to the default output device.

**Notes**

Use this function only with Ogg, Mp3, WMA, Mp4, Flac and Wav files. Other files will cause an exception.

Make sure that you stop playing a file before you de-init the IsoSDK. Otherwise you will get an exception.

## 1.2.164 PlayAudioTrack Function

**C++**

```
int32 BS_CALL PlayAudioTrack(int16 nTrackNumber);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int16 nTrackNumber	The tracknumber (index) you want to play.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will play a audio track on the current disc or disc image.

## 1.2.165 Prepare Function

**C++**

```
int32 BS_CALL Prepare();
```

**File**

IsoSDKExport.h ([see page 577](#))

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables.

**Notes**

This function maybe need some time to return if you manage many small files on the disc.

## 1.2.166 ReadCDText Function

**C++**

```
int32 BS_CALL ReadCDText(int32 index, int32 * pHandle);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 * pHandle	A handle to receive the information.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will create a handle to the CD-Text information of the audio disc according to the give device index.

## 1.2.167 ReadDirectory Function

**C++**

```
int32 BS_CALL ReadDirectory(HDIR hDirectory, int32 nIndex, struct SFileEntry* pFileEntry);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HDIR hDirectory	Handle to an open directory.
int32 nIndex	Entry position in the directory.
struct SFileEntry* pFileEntry	A pointer to a SFileEntry (see page 40) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function returns a pointer to a structure representing the directory entry at the specified position.

## 1.2.168 ReadFileContents Function

**C++**

```
int32 BS_CALL ReadFileContents(HSESSION hSession, const TCHAR* lpszFilePath, int64 nOffset,
void* pBuffer, int32 nBufferSize, int32* pRead);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	Handle of opened session.
const TCHAR* lpszFilePath	File path on disc (examples: "\file.bin", "\Dir1\Dir2\file.ext").
int64 nOffset	Offset in file.
void* pBuffer	A pointer to a output buffer.

int32 nBufferSize	Output buffer size in bytes = number of bytes to read.
int32* pRead	Number of bytes readed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function reads a segment of file from disc to a buffer.

## 1.2.169 ReadSectors Function

**C++**

```
int32 BS_CALL ReadSectors(int32 index, int32 nLBA, int32 nCount, int8 nFormat, void* pBuff, int32 nBuffLength);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nLBA	The current LBA to read.
int32 nCount	Number of tries to read.
int8 nFormat	Format to read. RAW (1) or normal (0).
void* pBuff	A pointer to the buffer you want to read to.
int32 nBuffLength	The length of the readed buffer. Possible are 2048 for default reading or 2352 for raw reading.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will read a given sector into a buffer.

## 1.2.170 ReleaseDeviceCapabilities Function

**C++**

```
int32 BS_CALL ReleaseDeviceCapabilities(int32 CapabilitiesHandle);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 CapabilitiesHandle	A handle that was created with GetDeviceCapabilitiesHandle (see page 162)().

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Release a created capability handle to clean up the memory.

## 1.2.171 RemoveBurnDevice Function

**C++**

```
int32 BS_CALL RemoveBurnDevice(const TCHAR* lpszDevice);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The device / drive letter of the device to be removed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will remove a device from the multi burn device list. This is important for multi-device burning.

**Notes**

Only the first letter or number of the device string will be used.

---

## 1.2.172 RemoveDir Function

**C++**

```
int32 BS_CALL RemoveDir(struct SDirToRemove DirToRemove);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SDirToRemove DirToRemove	A structure SDirToRemove (see page 21).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function removes a directory from the current project.

## 1.2.173 RemoveFile Function

### C++

```
int32 BS_CALL RemoveFile(struct SFileToRemove FileToRemove);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SFileToRemove FileToRemove	A structure SFileToRemove ( <a href="#">see page 53</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

### Conditions

This function removes a file from the project.

## 1.2.174 RenameDir Function

### C++

```
int32 BS_CALL RenameDir(struct SDirToRename DirToRename);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SDirToRename DirToRename	A structure SDirToRename ( <a href="#">see page 23</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function rename / move an directory in the project.

## 1.2.175 RenameFile Function

### C++

```
int32 BS_CALL RenameFile(struct SFileToRename FileToRename);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
struct SFileToRename FileToRename	A structure SFileToRename ( <a href="#">see page 54</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function rename / move a file in the project.

## 1.2.176 RescanDevices Function

**C++**

```
int32 BS_CALL RescanDevices();
```

**File**

IsoSDKExport.h ([see page 577](#))

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.

**Notes**

You can call this function with a timer to check if a new device is connected.

If you use the Windows DeviceChangeEvent you can call this function on this event.

Make sure that you call GetDevices ([see page 164](#)()) after this call.

## 1.2.177 ResetCallbacks Function

**C++**

```
int32 BS_CALL ResetCallbacks();
```

**File**

IsoSDKExport.h ([see page 577](#))

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will reset all initialized callbacks to your project.

**Notes**

If you call this function before the last callback was fired, a general bad crash will occur.

## 1.2.178 SaveLogToFile Function

### C++

```
int32 BS_CALL SaveLogToFile(const TCHAR* pchFilePath);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR* pchFilePath	Pointer to the file (with full path) to which you want to save the log data.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will save the automatically generated burning log data to a file.

### Notes

The IsoSDK will generate a text file if the file does not exist. This data is important for later support requests.

## 1.2.179 SaveOptionsToFile Function

### C++

```
int32 BS_CALL SaveOptionsToFile(const TCHAR * pchPath);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR * pchPath	Pointer of the XML file with full path to which you want to save the options.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will save the burn settings/ options to a XML File.

### Notes

If the XML file does not exist, the IsoSDK will create this file automatically. We recommend to store the file in your application folder.

## 1.2.180 SaveTrackToFile Function

### C++

```
int32 BS_CALL SaveTrackToFile(int32 index, int16 nTrackNumber, const TCHAR* pchFileName,
int32 nFileFormat);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int16 nTrackNumber	The number of the track inside the disc or image file.
const TCHAR* pchFileName	The path and name of the target file.
int32 nFileFormat	The fileformat of the track. Following values are possible: BS_FF_WAVE ( <a href="#">see page 336</a> ) BS_FF_ISO ( <a href="#">see page 336</a> ) BS_FF_BIN ( <a href="#">see page 335</a> ) BS_FF_MPEG ( <a href="#">see page 336</a> )

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function saves a selected track to a file.

## 1.2.181 SetAddFileEventCallback Function

### C++

```
int32 BS_CALL SetAddFileEventCallback(AddFileEvent AddFileEventCallback, void * pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
AddFileEvent AddFileEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Specifies an own function as a callback of the type AddFileEvent ([see page 235](#)). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef AddFileEvent
void CSampleBurnDlg::OnAddFileEvent(const TCHAR *pcFullPath, const TCHAR *pcJolietName,
const TCHAR *pcISOName, const TCHAR* pcUDFName, double dDateTime, double dFileSize,
CSSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetAddFileEventCallback((AddFileEvent) CSampleBurnDlg::OnAddFileEvent, this);
```

---

## 1.2.182 SetASPI Function

**C++**

```
int32 BS_CALL SetASPI(int32 nAspiInterface);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 nAspiInterface	With this parameter you can set the ASPI interface you want to use. The IsoSDK supports: BS_ASPI_INTERNAL ( <a href="#">see page 310</a> ) BS_ASPI_WNASPI ( <a href="#">see page 310</a> ) BS_ASPI_FROGASPI ( <a href="#">see page 309</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

If you want to use the internal ASPI layer after the initialization you use this function to switch from the external to the internal ASPI layer et vice versa .

**Notes**

After you have called this function you have to set the current project again. All properties and settings of the current project will be deleted. You can only use the internal ASPI layer with Windows 2000 or higher.

**Important:**

Never use a unknown or renamed ASPI layer that you found somewhere. This will definitely cause problems while redistribute your software. A public sample is the Nero WinAspi.dll (wnaspi32.dll). This can cause crashes or missfuntions on some systems with the use of another software than Nero.

Allways use the Internal Aspi for Windows 2000 and above. It is possible that a external ASPI layer will not detect new drives or suspend while burning.

## 1.2.183 SetAudioDecodeDoneEventCallback Function

### C++

```
int32 BS_CALL SetAudioDecodeDoneEventCallback( AudioDecodeDoneEvent AudioDecodeDoneCallback,  
void * pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
AudioDecodeDoneEvent AudioDecodeDoneCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Specifies an own function as a callback of the type AudioDecodeDoneEvent ([see page 236](#)). This function will be called every time the event triggers inside the IsoSDK.

### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

### Example

```
//A function according to the typedef AudioDecodeDoneEvent  
void CSampleBurnDlg::OnAudioDecodeDone(const TCHAR *pcFileName, const TCHAR *pcError, int32  
nErrorCode, CSampleBurnDlg *pUserData)  
{  
...  
}  
//Set the function as a callback to the SDK  
SetAudioDecodeDoneEventCallback((AudioDecodeDoneEvent)  
CSampleBurnDlg::OnAudioDecodeDoneEvent, this);
```

## 1.2.184 SetAudioDecoderEventCallback Function

### C++

```
int32 BS_CALL SetAudioDecoderEventCallback( AudioDecoderEvent AudioDecoderEventCallback,  
void * pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
AudioDecoderEvent AudioDecoderEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code

will be returned (see Error Codes).

#### Description

Specifies an own function as a callback of the type AudioDecoderEvent (see page 236). This function will be called every time the event triggers inside the IsoSDK.

#### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

#### Example

```
//A function according to the typedef AudioDecoderEvent
void CSampleBurnDlg::OnAudioDecoder(float fPercent, const TCHAR *pcFileName, int32 nType,
CSampleBurnDlg *pUserData)
{
...
}

//Set the function as a callback to the SDK
SetAudioDecoderEventCallback(AudioDecoderEvent) CSampleBurnDlg::OnAudioDecoderEvent, this);
```

## 1.2.185 SetAudioFileProperty Function

#### C++

```
int32 BS_CALL SetAudioFileProperty(struct SFileAudioProperty FileToAdd);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
struct SFileAudioProperty FileToAdd	A SFileAudioProperty (see page 34) structure.

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

The function sets the audio properties for the specified audio file in the project or for the disc. Supported projects: Audio, Mixed Mode.

## 1.2.186 SetBootInfoEx Function

#### C++

```
int32 BS_CALL SetBootInfoEx(struct SBootInfoEx BootInfoEx);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
struct SBootInfoEx BootInfoEx	A structure SBootInfoEx (see page 6).

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.

**Notes**

All settings are part of the “El Torito” specification. Please make sure that you are familiar with them in order to use the settings.

These settings will be reset after a new call of a data project.

## 1.2.187 SetBurnDevice Function

**C++**

```
int32 BS_CALL SetBurnDevice(const TCHAR* lpszDevice);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
const TCHAR* lpszDevice	The device/drive letter of the device to be selected.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the current device. This drive will be used for the burning process.

**Notes**

Only the first letter or number of the device string will be used.

## 1.2.188 SetBurnDoneEventCallback Function

**C++**

```
int32 BS_CALL SetBurnDoneEventCallback(BurnDoneEvent WriteDoneEventCallback, void * pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
BurnDoneEvent WriteDoneEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type BurnDoneEvent ([see page 237](#)). This function will be called every time

the event triggers inside the IsoSDK.

#### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

#### Example

```
//A function according to the typedef BurnDoneEvent
void CSampleBurnDlg::OnBurnDoneEvent(const TCHAR *pcError, CSampleBurnDlg *pUserData)
{
...
//Set the function as a callback to the SDK
SetBurnDoneEventCallback((BurnDoneEvent) CSampleBurnDlg::OnBurnDoneEvent, this);
```

## 1.2.189 SetBurnFileEventCallback Function

#### C++

```
int32 BS_CALL SetBurnFileEventCallback(BurnFileEvent BurnFileCallback, void * pUserData);
```

#### File

IsoSDKExport.h ([see page 577](#))

#### Parameters

Parameters	Description
BurnFileEvent BurnFileCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

#### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

Specifies an own function as a callback of the type BurnFileEvent ([see page 237](#)). This function will be called every time the event triggers inside the IsoSDK.

#### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

#### Example

```
//A function according to the typedef BurnFileEvent
void CSampleBurnDlg::OnBurnFileEvent(const TCHAR *pcFileName, CSampleBurnDlg *pUserData)
{
...
//Set the function as a callback to the SDK
SetBurnFileEventCallback((BurnFileEvent) CSampleBurnDlg::OnBurnFileEvent, this);
```

## 1.2.190 SetBurnSpeed Function

#### C++

```
int32 BS_CALL SetBurnSpeed(int32 index, int32 nSpeed);
```

#### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 nSpeed	New speed.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the writing speed of the current device.

**Notes**

If you use multiple devices (multi-device burning) this option will set the speed for all devices. If a device will not support the speed, it will be set to the nearest speed to this value.

You have to give a valid Speed value to this function. A valid Speed value is a value out of the GetPossibleBurnSpeeds (see page 176) function result.

Please note that there are different values for CD, DVD and Blu-Ray. The IsoSDK can only detect the right values if a blank media is inserted into the target drive.

If no blank media is available the IsoSDK will return the possible Speeds for CD-R.

## 1.2.191 SetCompareFilesForArrangementEventCallback Function

**C++**

```
int32 BS_CALL SetCompareFilesForArrangementEventCallback(CompareFilesForArrangementEvent
callback, void* pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
CompareFilesForArrangementEvent callback	Name of the event.
void* pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type CompareFilesForArrangementEvent (see page 237). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef CompareFilesForArrangementEvent
void CSampleBurnDlg::OnAddFileEvent(const TCHAR *pcFullPath, const TCHAR *pcJolietName,
const TCHAR *pcISOName, const TCHAR* pcUDFName, double dDateTime, double dFileSize,
CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetAddFileEventCallback((AddFileEvent) CSampleBurnDlg::OnAddFileEvent, this);
```

## 1.2.192 SetCompressEncrypt Function

**C++**

```
int32 BS_CALL SetCompressEncrypt(struct SCompressEncryptInfo CEInfo);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
struct SCompressEncryptInfo CEInfo	A structure SCompressEncryptInfo ( <a href="#">see page 11</a> ).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will set the parameters for internal compression and encryption.

## 1.2.193 SetCreateDirEventCallback Function

**C++**

```
int32 BS_CALL SetCreateDirEventCallback(CreateDirEvent CreateDirEventCallback, void * pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
CreateDirEvent CreateDirEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type CreateDirEvent ([see page 238](#)). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef CreateDirEvent
void CSampleBurnDlg::OnCreateDirEvent(const TCHAR *pcFullName, const TCHAR *pcJolietName,
const TCHAR* pcISOName, const TCHAR* pcUDFName, CSampleBurnDlg *pUserData)
{
...
}

//Set the function as a callback to the SDK
SetCreateDirEventCallback((CreateDirEvent) CSampleBurnDlg::OnCreateDirEvent, this);
```

---

## 1.2.194 SetDVDVideoOptions Function

**C++**

```
int32 BS_CALL SetDVDVideoOptions(struct SDVDVideoOptions DVDVideoOptions);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
struct SDVDVideoOptions DVDVideoOptions	A SDVDVideoOptions ( <a href="#">see page 28</a> ) structure.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the special options for VideoDVD projects.

---

## 1.2.195 SetEraseDoneEventCallback Function

**C++**

```
int32 BS_CALL SetEraseDoneEventCallback(EraseDoneEvent EraseDoneEventCallback, void * pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
EraseDoneEvent EraseDoneEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type EraseDoneEvent ([see page 238](#)). This function will be called every

time the event triggers inside the IsoSDK.

#### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

#### Example

```
//A function according to the typedef EraseDoneEvent
void CSampleBurnDlg::OnEraseDoneEvent(const TCHAR *pcError, CSampleBurnDlg *pUserData)
{
...
//Set the function as a callback to the SDK
SetEraseDoneEventCallback((EraseDoneEvent) CSampleBurnDlg::OnEraseDoneEvent, this);
```

---

## 1.2.196 SetFileAttr Function

#### C++

```
int32 BS_CALL SetFileAttr(const TCHAR * lpszFilePath, int32 nFileAttributes);
```

#### File

IsoSDKExport.h ([see page 577](#))

#### Parameters

Parameters	Description
const TCHAR * lpszFilePath	File name + full path of the file to the the attribute to.
int32 nFileAttributes	New attribute. Possible attributes: BS_FA_ARCHIVE ( <a href="#">see page 334</a> ) BS_FA_DIRECTORY ( <a href="#">see page 334</a> ) BS_FA_HIDDEN ( <a href="#">see page 334</a> ) BS_FA_ADVANCED_HIDDEN ( <a href="#">see page 333</a> ) BS_FA_READONLY ( <a href="#">see page 335</a> ) BS_FA_SYSTEM ( <a href="#">see page 335</a> )

#### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

This function set the attributes to a file.

---

## 1.2.197 SetFileTimeEx Function [new](#)

#### C++

```
int32 BS_CALL SetFileTimeEx(struct SFileTimeEx FileTimeEx);
```

#### File

IsoSDKExport.h ([see page 577](#))

#### Parameters

Parameters	Description
struct SFileTimeEx FileTimeEx	A structure SFileTimeEx ( <a href="#">see page 44</a> )

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.

You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes (see page 211) method.

This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.

## 1.2.198 SetFileTimes Function

**C++**

```
int32 BS_CALL SetFileTimes(const TCHAR * lpszFilePath, const SFileDateTime * pCreationTime,
                           const SFileDateTime * pModificationTime, const SFileDateTime * pAccessTime);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR * lpszFilePath	File name + full path of the file to the the value to.
const SFileDateTime * pCreationTime	Creation time of the file.
const SFileDateTime * pModificationTime	Modification time of the file.
const SFileDateTime * pAccessTime	Last access time of the file.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will set the custom dates to a file.

**Notes**

If you do not set this values, the IsoSDK will use the common information from the files system.

## 1.2.199 SetFileUserParam Function

**C++**

```
int32 BS_CALL SetFileUserParam(const TCHAR * lpszFilePath, void * pData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR * lpszFilePath	File name + full path of the file to the the value to.

void * pUserData	A pointer to a value or custom structure.
------------------	---

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.

## 1.2.200 SetFinalizeEventCallback Function

#### C++

```
int32 BS_CALL SetFinalizeEventCallback(FinalizeEvent FinalizeEventCallback, void * pUserData);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
FinalizeEvent FinalizeEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

Specifies an own function as a callback of the type FinalizeEvent (see page 239). This function will be called every time the event triggers inside the IsoSDK.

#### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

#### Example

```
//A function according to the typedef FinalizeEvent
void CSampleBurnDlg::OnFinalizeEvent(CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetFinalizeEventCallback((FinalizeEvent) CSampleBurnDlg::OnFinalizeEvent, this);
```

## 1.2.201 SetFXApp Function

#### C++

```
int32 BS_CALL SetFXApp(void* pApp);
```

#### File

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
void* pApp	FoxToolkit handle.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Set the Fox-Toolkit App handle to the SDK.

## 1.2.202 SetGetTextEventCallback Function

**C++**

```
int32 BS_CALL SetGetTextEventCallback(GetTextEvent GetTextEventCallback, void * pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
GetTextEvent GetTextEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type GetTextEvent (see page 239). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef GetTextEvent
void CSampleBurnDlg::OnGetTextEvent(int32 nTextID, TCHAR *pchText, int32 *pnLength,
CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetGetTextEventCallback((GetTextEvent) CSampleBurnDlg::OnGetTextEvent, this);
```

## 1.2.203 SetIgnoreFileExist Function new

**C++**

```
int32 BS_CALL SetIgnoreFileExist(BS_BOOL bIgnore);
```

**File**

IsoSDKExport.h (see page 577)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

The IsoSDK will check about already added files while AddFile (see page 119) and AddDir (see page 118) function. If you have industrial workflows with thousands of files you can disable this check with this function.

**Notes**

While project type change or reset this value will get set to default (off).

## 1.2.204 SetImagePath Function

**C++**

```
int32 BS_CALL SetImagePath(const TCHAR* pchPath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR* pchPath	File name + full path of the image file.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

The function sets the path to the image (\*.iso) file you want to burn using the 'Imagewriter'.

## 1.2.205 SetInfoTextEventCallback Function

**C++**

```
int32 BS_CALL SetInfoTextEventCallback(InfoTextEvent InfoTextCallback, int32 nLevel, void * pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
InfoTextEvent InfoTextCallback	Name of the event.
int32 nLevel	Information priority (Level): BS_IL_INFO (see page 356) BS_IL_LOW_DEBUG (see page 357) BS_IL_MEDIUM_DEBUG (see page 357) BS_IL_HIGH_DEBUG (see page 356)
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type InfoTextEvent (see page 240). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef InfoTextEvent
void CSampleBurnDlg::OnInfoTextEvent(const TCHAR *pcInfoText, int32 nLevel, CSampleBurnDlg
*pUserData)
{
...
}

//Set the function as a callback to the SDK
SetInfoTextEventCallback((InfoTextEvent) CSampleBurnDlg::OnInfoTextEvent, BS_IL_INFO, this);
```

## 1.2.206 SetISOInfoEx Function

**C++**

```
int32 BS_CALL SetISOInfoEx(struct SISOInfoEx IsoInfoEx);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
struct SISOInfoEx IsoInfoEx	A structure SISOInfoEx (see page 56).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to set the extended ISO Fields for Data CD/DVD and ISO images. Available only in data projects.

**Notes**

Very important: Since Joliet is 2 Bytes, the max. usable length of the strings in this structure is only the reservation length divided by 2. Otherwise your strings will be cut.

These settings will be reset after a new call of a data project.

## 1.2.207 SetJobDoneEventCallback Function

**C++**

```
int32 BS_CALL SetJobDoneEventCallback(JobDoneEvent JobDoneEventCallback, void * pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
JobDoneEvent JobDoneEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type JobDoneEvent (see page 241). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef JobDoneEvent
void CSampleBurnDlg::OnJobDoneEvent(CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetJobDoneEventCallback((JobDoneEvent) CSampleBurnDlg::OnJobDoneEvent, this);
```

## 1.2.208 SetLanguage Function

**C++**

```
int32 BS_CALL SetLanguage(const TCHAR * pchLanguage, const TCHAR * pchLanguageFile);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
const TCHAR * pchLanguage	Language. This is the language section in the INI file (e.g. [English]).
const TCHAR * pchLanguageFile	File name of the language ini file (if you do not provide a folder, the IsoSDK will look for the specified file in the program folder).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the language of the IsoSDK. The following resources are effected: error codes, message codes and GUI codes.

**Notes**

Set the language of the IsoSDK as early as possible.

## 1.2.209 SetOptions Function

### C++

```
int32 BS_CALL SetOptions(struct SOptions Options);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SOptions Options	A structure SOptions ( <a href="#">see page 75</a> ).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function to set the burning options.

## 1.2.210 SetOptionsFromFile Function

### C++

```
int32 BS_CALL SetOptionsFromFile(const TCHAR * pchPath);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
<b>const</b> TCHAR * pchPath	Pointer of the XML file with full path from which you want to set the options.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function will set the burn settings/ options from a previously stored XML file.

### Notes

Please use only XML files that were stored by the IsoSDK.

## 1.2.211 SetProcessEventCallback Function

### C++

```
int32 BS_CALL SetProcessEventCallback(ProcessEvent ProcessCallback, void * pUserData);
```

### File

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
ProcessEvent ProcessCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type ProcessEvent (see page 242). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef ProcessEvent
void CSampleBurnDlg::OnProcessEvent(float fPercent, float fDeviceBuffer, float fCache,
double dBytesWritten, double dImageSize, CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetProcessEventCallback((ProcessEvent) CSampleBurnDlg::OnProcessEvent, this);
```

## 1.2.212 SetRAWDataEventCallback Function

**C++**

```
int32 BS_CALL SetRAWDataEventCallback(RAWDataEvent RAWDataEventCallback, void* pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
RAWDataEvent RAWDataEventCallback	Name of the event.
void* pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type RAWDataEvent (see page 242). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef RAWDataEvent
void CSampleBurnDlg::OnRAWDataEvent(int32 nLBA, void* pBuffer, int32 nBufferSize, int32*
nReaded, CSampleBurnDlg *pUserData)
```

```
{
...
}
//Set the function as a callback to the SDK
SetRAWDataEventCallback((RAWDataEvent) CSampleBurnDlg::OnRAWDataEvent, this);
```

## 1.2.213 SetRAWStructure Function

### C++

```
int32 BS_CALL SetRAWStructure(struct SRAWTrackToAdd* pTracks, int32 nLength);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
struct SRAWTrackToAdd* pTracks	A pointer to a SRAWTrackToAdd ( <a href="#">see page 82</a> ) structure.
int32 nLength	Number of tracks inside pTracks.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

The function sets the structure of the disc and the format of the input data for a RAW project.

## 1.2.214 SetReadDevice Function

### C++

```
int32 BS_CALL SetReadDevice(const TCHAR* lpszDevice);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
const TCHAR* lpszDevice	The device/drive letter of the device to be selected.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function sets the reading speed of the current read device.

## 1.2.215 SetReadSpeed Function

### C++

```
int32 BS_CALL SetReadSpeed(int32 index, int32 nSpeed);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
int32 index	The index of the device you want to get the information from. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int32 nSpeed	The new read speed value. If you only want to use the max. reading speed, pass the value BS_MAX_SPEED ( <a href="#">see page 364</a> ). This is the default value.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function sets the reading speed of the current device.

**Notes**

You have to give a valid Speed value to this function. A valid Speed value is a value out of the GetPossibleReadSpeeds ([see page 177](#)) function result.

Please note that there are different values for CD, DVD and Blu-Ray. The IsoSDK can only detect the right values if a blank media is inserted into the target drive.

If no blank media is available the IsoSDK will return the possible Speeds for CD-R.

---

## 1.2.216 SetRegionalCode Function

**C++**

```
int32 BS_CALL SetRegionalCode(int32 index, int32 nRPC);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
int32 index	The index of the device you want to use to set the new region code. Call GetActiveDevicesCount ( <a href="#">see page 155</a> ()) to get the device count.
int32 nRPC	The new region code to be set.

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0.

**Notes**

The change of a region code is limited! So if you allow this function in your software make sure to inform the user about this. You can get the "changes left" with the function GetDeviceInformationEx ([see page 163](#)()).

## 1.2.217 SetRemoveFileEventCallback Function

### C++

```
int32 BS_CALL SetRemoveFileEventCallback(RemoveFileEvent RemoveFileEventCallback, void * pUserData);
```

### File

IsoSDKEExport.h ([see page 577](#))

### Parameters

Parameters	Description
RemoveFileEvent RemoveFileEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Specifies an own function as a callback of the type RemoveFileEvent ([see page 243](#)). This function will be called every time the event triggers inside the IsoSDK.

### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

### Example

```
//A function according to the typedef RemoveFileEvent
void CSampleBurnDlg::OnRemoveFileEvent(const TCHAR *pcFullPath, const TCHAR
*pcDestinationPath, const TCHAR *pcFile, CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetRemoveFileEventCallback((RemoveFileEvent) CSampleBurnDlg::OnRemoveFileEvent, this);
```

## 1.2.218 SetStartVerifyEventCallback Function

### C++

```
int32 BS_CALL SetStartVerifyEventCallback(StartVerifyEvent StartVerifyEventCallback, void * pUserData);
```

### File

IsoSDKEExport.h ([see page 577](#))

### Parameters

Parameters	Description
StartVerifyEvent StartVerifyEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

## Description

Specifies an own function as a callback of the type StartVerifyEvent (see page 243). This function will be called every time the event triggers inside the IsoSDK.

## Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

## Example

```
//A function according to the typedef StartVerifyEvent
void CSampleBurnDlg::OnStartVerifyEvent(CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetStartVerifyEventCallback((StartVerifyEvent) CSampleBurnDlg::OnStartVerifyEvent, this);
```

## 1.2.219 SetTmpPath Function

### C++

```
int32 BS_CALL SetTmpPath(const TCHAR * pchPath);
```

### File

IsoSDKExport.h (see page 577)

### Parameters

Parameters	Description
const TCHAR * pchPath	This value contains the desired path (this must be an already existing path). The value must be null terminated.

### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

## Description

With the help of this function the temporary folder is set. A previously set path will be overwritten.

### Notes

Make the change to the path as early as possible. If you call this function later, the current project may be reset.

## 1.2.220 SetUDFOptions Function

### C++

```
int32 BS_CALL SetUDFOptions(struct SUDFOptions UdfOptions);
```

### File

IsoSDKExport.h (see page 577)

### Parameters

Parameters	Description
struct SUDFOptions UdfOptions	A structure SUDFOptions (see page 97).

### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code

will be returned (see Error Codes).

#### Description

Use this function to set the UDF options for UDF data projects.

## 1.2.221 SetUDFOptionsEx Function

#### C++

```
int32 BS_CALL SetUDFOptionsEx(struct SUDFOptionsEx * pUdfOptions);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
<code>struct SUDFOptionsEx * pUdfOptions</code>	A structure SUDFOptionsEx (see page 232).

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

Use this function to set the UDF extended options for UDF data projects. Insert additional UDF settings with this function.

## 1.2.222 SetVCDKeyHandler Function

#### C++

```
int32 BS_CALL SetVCDKeyHandler(int32 nItemNumber, int32 nKey, int32 nNextItemNumber);
```

#### File

IsoSDKExport.h (see page 577)

#### Parameters

Parameters	Description
<code>int32 nItemNumber</code>	Item number to set the key handler to.
<code>int32 nKey</code>	ID of the pressed key. Following IDs are allowed: <code>BS_VCD_KEY_0</code> (see page 504), <code>BS_VCD_KEY_DEFAULT</code> (see page 504), <code>BS_VCD_KEY_RETURN</code> (see page 505), <code>BS_VCD_KEY_PREVIOUS</code> (see page 505), <code>BS_VCD_KEY_NEXT</code> (see page 505)
<code>int32 nNextItemNumber</code>	Item number of the next item.

#### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

#### Description

This function sets which item will be shown next when particular key is pressed.

## 1.2.223 SetVCDTimeOutHandler Function

### C++

```
int32 BS_CALL SetVCDTimeOutHandler(int32 nItemNumber, int32 nTimeOut, int32  
  nNextItemNumber);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
int32 nItemNumber	Item number to set the timeout to.
int32 nTimeOut	Timeout value.
int32 nNextItemNumber	Item number of the next item.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

This function sets transition timeout between items.

## 1.2.224 SetVerify Function

### C++

```
int32 BS_CALL SetVerify(BS_BOOL bVerify);
```

### File

IsoSDKExport.h ([see page 577](#))

### Parameters

Parameters	Description
BS_BOOL bVerify	Pass BS_TRUE ( <a href="#">see page 499</a> ) to activate afterburn verification. Pass BS_FALSE ( <a href="#">see page 335</a> ) to burn without verification.

### Returns

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Use this function to set the after burn verification.

## 1.2.225 SetVerifyDoneEventCallback Function

### C++

```
int32 BS_CALL SetVerifyDoneEventCallback(VerifyDoneEvent VerifyDoneEventCallback, void *  
  pUserData);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
VerifyDoneEvent VerifyDoneEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type VerifyDoneEvent ([see page 244](#)). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef VerifyDoneEvent
void CSampleBurnDlg::OnVerifyDoneEvent(int32 nNumErrors, CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetVerifyDoneEventCallback((VerifyDoneEvent) CSampleBurnDlg::OnVerifyDoneEvent, this);
```

## 1.2.226 SetVerifyErrorEventCallback Function

**C++**

```
int32 BS_CALL SetVerifyErrorEventCallback(VerifyErrorEvent VerifyErrorEventCallback, void * pUserData);
```

**File**IsoSDKExport.h ([see page 577](#))**Parameters**

Parameters	Description
VerifyErrorEvent VerifyErrorEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type VerifyErrorEvent ([see page 245](#)). This function will be called every time the event triggers inside the SDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef VerifyErrorEvent
void CSampleBurnDlg::OnVerifyErrorEvent(const TCHAR *pcFileName, const TCHAR *pcError,
CSampleBurnDlg *pUserData)
{
...
}

//Set the function as a callback to the SDK
SetVerifyErrorEventCallback((VerifyErrorEvent) CSampleBurnDlg::OnVerifyErrorEvent, this);
```

---

## 1.2.227 SetVerifyFileEventCallback Function

**C++**

```
int32 BS_CALL SetVerifyFileEventCallback(VerifyFileEvent VerifyFileEventCallback, void *
pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
VerifyFileEvent VerifyFileEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO ([see page 469](#)) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type VerifyFileEvent ([see page 245](#)). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef VerifyFileEvent
void CSampleBurnDlg::OnVerifyFileEvent(const TCHAR *pcFileName, CSampleBurnDlg *pUserData)
{
...

}

//Set the function as a callback to the SDK
SetVerifyFileEventCallback((VerifyFileEvent) CSampleBurnDlg::OnVerifyFileEvent, this);
```

---

## 1.2.228 SetVerifySectorEventCallback Function

**C++**

```
int32 BS_CALL SetVerifySectorEventCallback(VerifySectorEvent VerifySectorEventCallback,
void * pUserData);
```

**File**

IsoSDKExport.h ([see page 577](#))

**Parameters**

Parameters	Description
VerifySectorEvent VerifySectorEventCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type VerifySectorEvent (see page 245). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef VerifySectorEvent
void CSampleBurnDlg::OnVerifySectorEvent(const TCHAR *pcFullPath, const TCHAR
*pcJolietName, const TCHAR *pcISOName, const TCHAR* pcUDFName , double dDateTime, double
dFileSize, CSampleBurnDlg *pUserData)
{
    ...
}
//Set the function as a callback to the SDK
SetVerifySectorEventCallback((VerifySectorEvent) CSampleBurnDlg::OnVerifySectorEvent, this);
```

## 1.2.229 SetVideoScanDoneEventCallback Function

**C++**

```
int32 BS_CALL SetVideoScanDoneEventCallback(VideoScanDoneEvent VideoScanDoneCallback, void
* pUserData);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
VideoScanDoneEvent VideoScanDoneCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Specifies an own function as a callback of the type VideoScanDoneEvent (see page 246). This function will be called every time the event triggers inside the IsoSDK.

**Notes**

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

**Example**

```
//A function according to the typedef VideoScanDoneEvent
```

```
void CSampleBurnDlg::OnVideoScanDoneEvent( const TCHAR *pcFileName, const TCHAR *pcError,
int32 nErrorCode, struct SVideoFormat CurrentFormat, CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetVideoScanDoneEventCallback((VideoScanDoneEvent) CSampleBurnDlg::OnVideoScanDoneEvent,
this);
```

## 1.2.230 SetVideoScannerEventCallback Function

### C++

```
int32 BS_CALL SetVideoScannerEventCallback(VideoScannerEvent VideoScannerCallback, void *
pUserData);
```

### File

IsoSDKExport.h (see page 577)

### Parameters

Parameters	Description
VideoScannerEvent VideoScannerCallback	Name of the event.
void * pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

### Returns

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

### Description

Specifies an own function as a callback of the type VideoScannerEvent (see page 246). This function will be called every time the event triggers inside the IsoSDK.

### Notes

Set the call backs as early as possible. They cannot be set during running processes (deletion, writing, scanning).

### Example

```
//A function according to the typedef VideoScannerEvent
void CSampleBurnDlg::OnVideoScannerEvent(float fPercent, const TCHAR *pcFileName,
CSampleBurnDlg *pUserData)
{
...
}
//Set the function as a callback to the SDK
SetVideoScannerEventCallback((VideoScannerEvent) CSampleBurnDlg::OnVideoScannerEvent, this);
```

## 1.2.231 SetWriteCDTextInUnicode Function

### C++

```
int32 BS_CALL SetWriteCDTextInUnicode(BS_BOOL bWriteInUnicode = BS_FALSE);
```

### File

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
BS_BOOL bWriteInUnicode = BS_FALSE	A BOOLEAN value. BS_TRUE (see page 499) the SDK will write ANSI, BS_FALSE (see page 335) the SDK will write in Unicode encoding.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

Use this function to set the CD-Text Unicode switch.

**Notes**

CD-Text itself is defined without Unicode. Some software products (players e.g) support Unicode CD-Text. But most players, plugins and whatever did not support Unicode.

## 1.2.232 StopMpegAction Function

**C++**

```
int32 BS_CALL StopMpegAction();
```

**File**

IsoSDKExport.h (see page 577)

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function interupts the current scanning process of an added video file to a VCD or SVCD project.

**Notes**

This function immediately stops the scan. No data is available for burning.

## 1.2.233 TagsFromNetworkDialog Function

**C++**

```
int32 BS_CALL TagsFromNetworkDialog(int32 nDeviceIndex, int32 * pHandle);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
int32 nDeviceIndex	The index of the device you want to get the information from. Call GetActiveDevicesCount (see page 155)() to get the device count.
int32 * pHandle	A handle to receive the information.

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function will create a handle to the CDDB information of the audio disc according to the give device index.

## 1.2.234 VerifyFile Function

**C++**

```
int32 BS_CALL VerifyFile(HSESSION hSession, const TCHAR* lpszSourcePath);
```

**File**

IsoSDKExport.h (see page 577)

**Parameters**

Parameters	Description
HSESSION hSession	The session where you want to receive the information from.
const TCHAR* lpszSourcePath	The path to the file or a path to a folder on the disc.. Example "\data\testfile.txt".

**Returns**

The function returns BS\_SDK\_ERROR\_NO (see page 469) upon successful execution. In case of an error the error code will be returned (see Error Codes).

**Description**

This function verify a file of the medium against whether it is is readable without any error.

## 1.3 Structs, Records, Enums

The following table lists structs, records, enums in this documentation.

**Structures**

	Name	Description
◆	_FoxEntries (see page 231)	This is record _FoxEntries.
◆	_FoxIndecies (see page 231)	This is record _FoxIndecies.
◆	SExtent (see page 231)	A structure that contains the information about the Extent information of a file in allocation table.
◆	SFileAllocationTable (see page 232)	A structure that contains the information about the file allocation table.
◆	SUDFOptionsEx (see page 232)	This structure is used to get and set the extended parameters for UDF disc. Added StructSize fields to the structure. This field must be filled-in with structure size by the caller. For example:
	FoxEntries (see page 233)	This is type FoxEntries.
	FoxIndecies (see page 233)	This is type FoxIndecies.
	PFoxEntries (see page 234)	This is type PFoxEntries.
	PFoxIndecies (see page 234)	This is type PFoxIndecies.

## 1.3.1 \_FoxEntries Structure

### C++

```
struct _FoxEntries {
    int64 lnOffset;
    int64 lnSize;
};
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

This is record \_FoxEntries.

## 1.3.2 \_FoxIndecies Structure

### C++

```
struct _FoxIndecies {
    int64 lnFileSize;
    unsigned long nSize;
    unsigned long nSignature;
};
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

This is record \_FoxIndecies.

## 1.3.3 SExtent Structure

### C++

```
struct SExtent {
    uint32 Location;
    uint32 Length;
};
```

### File

IsoSDKDefinitions.h (see page 517)

### Members

Members	Description
uint32 Location;	LBA of the first sector of the file extent
uint32 Length;	Size of the file extent in bytes

### Description

A structure that contains the information about the Extent information of a file in allocation table.

## 1.3.4 SFileAllocationTable Structure

### C++

```
struct SFileAllocationTable {
    int16 EmbeddedFileOffset;
    int8 NumExtents;
    SExtent Extents[1];
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Members

Members	Description
int16 EmbeddedFileOffset;	If it is not 0 it contains the offset in bytes of the file data in File Entry (file descriptor) and it will have exact one extend info with the file size and location. If this is 0 there will be an array of extents available
int8 NumExtents;	Number/index of SExtend in the Extends array
SExtent Extents[1];	An array with the available SExtent ( <a href="#">see page 231</a> ) information

### Description

A structure that contains the information about the file allocation table.

## 1.3.5 SUDFOptionsEx Structure

### C++

```
struct SUDFOptionsEx {
    int32 StructSize;
    int32 Mask;
    int32 Version;
    int32 PartitionType;
    BS_BOOL WriteFileStreams;
    BS_BOOL AvchdDisc;
    char ImplementationID[23];
};
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Members

Members	Description
int32 StructSize;	Size of the current used / to use structure.
int32 Mask;	The mask to set the option you want to query / use with the SUDFOptionsEx.
int32 Version;	UDF version. Possible values are: BS_UDF_VERSION_102 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_150 ( <a href="#">see page 500</a> ), BS_UDF_VERSION_200 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_201 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_250 ( <a href="#">see page 501</a> ), BS_UDF_VERSION_260 ( <a href="#">see page 501</a> )

int32 PartitionType;	UDF partition type. Possible Values are: BS_UDF_PARTITION_PHYSICAL (see page 499), BS_UDF_PARTITION_VIRTUAL (see page 500), BS_UDF_PARTITION_SPARABLE (see page 499)
BS_BOOL WriteFileStreams;	States whether file streams should be used.
BS_BOOL AvchdDisc;	Allow to burn the Blu-ray file system to a DVD blank.
char ImplementationID[23];	This is the volume name of a UDF file system disc.

**Description**

This structure is used to get and set the extended parameters for UDF disc. Added StructSize fields to the structure. This field must be filled-in with structure size by the caller. For example:

```
SUDFOptionsEx options;
options.StructSize = sizeof(SUDFOptionsEx);
```

Also these functions allow to get and set UDF options selectively, not the whole at once as before.

**Example**

For example to set the UDF ImplementationID without touching all other options

```
SUDFOptionsEx options;
memset(&options, 0, sizeof(options));
options.StructSize = sizeof(options);
options.Mask = BS_UDFOPT_IMPLEMENTATION_ID;
strncpy(options.ImplementationID, "SampleImpl", sizeof(options.ImplementationID));
SetUDFOptionsEx(&options);
```

Or

```
SUDFOptionsEx options = { sizeof(SUDFOptionsEx), BS_UDFOPT_IMPLEMENTATION_ID };
strncpy(options.ImplementationID, "SampleImpl", sizeof(options.ImplementationID));
SetUDFOptionsEx(&options);
```

---

## 1.3.6 FoxEntries Structure

**C++**

```
typedef struct _FoxEntries {
    int64 lnOffset;
    int64 lnSize;
} FoxEntries, * PFoxEntries;
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is type FoxEntries.

---

## 1.3.7 FoxIndecies Structure

**C++**

```
typedef struct _FoxIndecies {
    int64 lnFileSize;
    unsigned long nSize;
    unsigned long nSignature;
} FoxIndecies, * PFoxIndecies;
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is type PFoxEntries.

## 1.3.8 PFoxEntries Structure

**C++**

```
typedef struct _FoxEntries {
    int64  lnOffset;
    int64  lnSize;
} FoxEntries, * PFoxEntries;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type PFoxEntries.

## 1.3.9 PFoxIndecies Structure

**C++**

```
typedef struct _FoxIndecies {
    int64  lnFileSize;
    unsigned long nSize;
    unsigned long nSignature;
} FoxIndecies, * PFoxIndecies;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type PFoxIndecies.

# 1.4 Types

The following table lists types in this documentation.

**Types**

Name	Description
AddFileEvent ( <a href="#">see page 235</a> )	This event will be triggered when a file was successfully added to the project.
AudioDecodeDoneEvent ( <a href="#">see page 236</a> )	This event will be triggered after a audio file was decoded.
AudioDecoderEvent ( <a href="#">see page 236</a> )	This event will be triggered during the decoding of a audio file. It shows the current progress of the process.
BurnDoneEvent ( <a href="#">see page 237</a> )	This event will be triggered when the burning process was completed.
BurnFileEvent ( <a href="#">see page 237</a> )	This event will be triggered for each file in the current burning process.
CompareFilesForArrangementEvent ( <a href="#">see page 237</a> )	This event will be triggered after a new dir was create in the current project.

CreateDirEvent ( <a href="#">see page 238</a> )	This event will be triggered after a new dir was create in the current project.
EraseDoneEvent ( <a href="#">see page 238</a> )	This event will be triggered when the deletion process was completed.
FinalizeEvent ( <a href="#">see page 239</a> )	This event will be triggered during the burning process when the finalize process starts. After this event is triggered the Event ProcessEvent ( <a href="#">see page 242</a> ) will report the progress of the finalize process.
GetTextEvent ( <a href="#">see page 239</a> )	This event will be triggered when you call the GetText ( <a href="#">see page 181</a> ()) function.
HDIR ( <a href="#">see page 239</a> )	HDIR is a virtual directory's handle. It is needed to read contents of disc session and IsoSDK's project. HDIR handles are obtained using OpenDirectory ( <a href="#">see page 192</a> ()) function. After finished using HDIR handle it should be closed by CloseDirectory ( <a href="#">see page 124</a> ()) function.
HSESSION ( <a href="#">see page 240</a> )	HSESSION is a virtual session handle. It is needed to list disc session and IsoSDK's projects contents, to read file system information, to import, read and verify files from disc. It is returned by OpenDiskSession ( <a href="#">see page 193</a> ()) function. After finished using HSESSION handle it should be closed by CloseDiskSession ( <a href="#">see page 125</a> ()).
InfoTextEvent ( <a href="#">see page 240</a> )	This event is triggered when the IsoSDK was loaded and an action was performed. This event is useable for logging.
int16 ( <a href="#">see page 240</a> )	This is type int16.
int32 ( <a href="#">see page 241</a> )	This is type int32.
int64 ( <a href="#">see page 241</a> )	This is type int64.
int8 ( <a href="#">see page 241</a> )	This is type int8.
JobDoneEvent ( <a href="#">see page 241</a> )	This event will be triggered when all operations with the device are finished (after burn or erase process) and the IsoSDK is available for further use.
ProcessEvent ( <a href="#">see page 242</a> )	This event will be triggered during a deletion or burn, erase and finalize process. It shows the current progress of the process.
RAWDataEvent ( <a href="#">see page 242</a> )	This event will be triggered if the IsoSDK needs more data for burning the RAW Image file. This event will only occur if no file was added to a BS_BRUNER_RAW project.
RemoveFileEvent ( <a href="#">see page 243</a> )	This event will be triggered when a file was removed from the project.
StartVerifyEvent ( <a href="#">see page 243</a> )	This event will be triggered when the verify process will start.
uint16 ( <a href="#">see page 243</a> )	This is type uint16.
uint32 ( <a href="#">see page 244</a> )	This is type uint32.
uint8 ( <a href="#">see page 244</a> )	This is type uint8.
VerifyDoneEvent ( <a href="#">see page 244</a> )	This event will be triggered when the verify process is finished.
VerifyErrorEvent ( <a href="#">see page 245</a> )	This event is triggered if an error occurred during the current verify process.
VerifyFileEvent ( <a href="#">see page 245</a> )	This event will be triggered every time a new file will be verified.
VerifySectorEvent ( <a href="#">see page 245</a> )	This event will be triggered every time a new file will be verified.
VideoScanDoneEvent ( <a href="#">see page 246</a> )	This event will be triggered when the video scanning process is completed.
VideoScannerEvent ( <a href="#">see page 246</a> )	This event will be triggered during a running scanning process. It shows the current progress of the process.

## 1.4.1 AddFileEvent Type

### C++

```
typedef void (* AddFileEvent)(const TCHAR* pcFullPath, const TCHAR* pcJolietName, const
TCHAR* pcISOName, const TCHAR* pcUDFName, double dDateTime, double dFileSize, void*
pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFullPath	Source path of the added file (includes file name).
pcJolietName	Joliet name of the added file.
pclISOName	ISO9660 name of the added file.
pcUDFName	UDF name of the added file.
dDateTime	Timestamp of the last change.
dFileSize	File size in bytes.
pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Description**

This event will be triggered when a file was successfully added to the project.

## 1.4.2 AudioDecodeDoneEvent Type

**C++**

```
typedef void (* AudioDecodeDoneEvent)(const TCHAR* pcFileName, const TCHAR* pcError, int32 nErrorCode, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFileName	Path of the decoded file (includes file name)
pcError	Burning error. If no error occurred, an empty string or a null pointer will be returned
nErrorCode	The ID of the Error according to the Errorcodes
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered after a audio file was decoded.

## 1.4.3 AudioDecoderEvent Type

**C++**

```
typedef void (* AudioDecoderEvent)(float fPercent, const TCHAR* pcFileName, int32 nType, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
fPercent	Progress in % (0 – 100)

pcFileName	Path of the file decoded (includes file name)
nType	*not used anymore*
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered during the decoding of a audio file. It shows the current progress of the process.

## 1.4.4 BurnDoneEvent Type

**C++**

```
typedef void (* BurnDoneEvent)(const TCHAR* pcError, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcError	Burning error. If no error occurred, an empty string or a null pointer will be returned.
pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Description**

This event will be triggered when the burning process was completed.

---

## 1.4.5 BurnFileEvent Type

**C++**

```
typedef void (* BurnFileEvent)(const TCHAR* pcFileName, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFileName	Source path of the burned file (includes file name).
pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Description**

This event will be triggered for each file in the current burning process.

---

## 1.4.6 CompareFilesForArrangementEvent Type

**C++**

```
typedef BS_BOOL (* CompareFilesForArrangementEvent)(const SFileEntry* pFile1, const
SFileEntry* pFile2, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFullName	The original name of the created directory
pcJolietName	Joliet name of the created directory
pclISOName	ISO9660 name of the created directory
pcUDFName	UDF name of the created directory
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered after a new dir was create in the current project.

## 1.4.7 CreateDirEvent Type

**C++**

```
typedef void (* CreateDirEvent)(const TCHAR* pcFullName, const TCHAR* pcJolietName, const
TCHAR* pclISOName, const TCHAR* pcUDFName, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFullName	The original name of the created directory
pcJolietName	Joliet name of the created directory
pclISOName	ISO9660 name of the created directory
pcUDFName	UDF name of the created directory
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered after a new dir was create in the current project.

## 1.4.8 EraseDoneEvent Type

**C++**

```
typedef void (* EraseDoneEvent)(const TCHAR* pcError, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcError	Deletion error. If no error occurred, an empty string or a null pointer will be returned
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when the deletion process was completed.

## 1.4.9 FinalizeEvent Type

**C++**

```
typedef void (* FinalizeEvent)(void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered during the burning process when the finalize process starts. After this event is triggered the Event ProcessEvent ([see page 242](#)) will report the progress of the finalize process.

## 1.4.10 GetTextEvent Type

**C++**

```
typedef int32 (* GetTextEvent)(int32 nTextID, TCHAR* pchText, int32* pnLength, void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
nTextID	This value is the Text-ID of the required text
pchText	This is corresponding text of the reported ID
pnLength	This is the text length of pchText
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when you call the GetText ([see page 181](#))() function.

## 1.4.11 HDIR Type

**C++**

```
typedef void* HDIR;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

HDIR is a virtual directory's handle. It is needed to read contents of disc session and IsoSDK's project. HDIR handles are obtained using OpenDirectory (see page 192)() function. After finished using HDIR handle it should be closed by CloseDirectory (see page 124)() function.

## 1.4.12 HSESSION Type

**C++**

```
typedef void* HSESSION;
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

HSESSION is a virtual session handle. It is needed to list disc session and IsoSDK's projects contents, to read file system information, to import, read and verify files from disc. It is returned by OpenDiskSession (see page 193)() function. After finished using HSESSION handle it should be closed by CloseDiskSession (see page 125)().

## 1.4.13 InfoTextEvent Type

**C++**

```
typedef void (* InfoTextEvent)(const TCHAR* pcInfoText, int32 nLevel, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcInfoText	Extended information about the current status of the IsoSDK
nLevel	This value states the information priority. BS_IL_INFO (see page 356) BS_IL_LOW_DEBUG (see page 357) BS_IL_MEDIUM_DEBUG (see page 357) BS_IL_HIGH_DEBUG (see page 356)
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event is triggered when the IsoSDK was loaded and an action was performed. This event is useable for logging.

## 1.4.14 int16 Type

**C++**

```
typedef signed short int16;
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is type int16.

---

## 1.4.15 int32 Type

**C++**

```
typedef signed int int32;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type int32.

---

## 1.4.16 int64 Type

**C++**

```
typedef long long int int64;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type int64.

---

## 1.4.17 int8 Type

**C++**

```
typedef signed char int8;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type int8.

---

## 1.4.18 JobDoneEvent Type

**C++**

```
typedef void (* JobDoneEvent)(void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

---

**Parameters**

Parameters	Description
pUserData	This value is a custom defined pointer (e.g. the pointer to the window).

**Description**

This event will be triggered when all operations with the device are finished (after burn or erase process) and the IsoSDK is available for further use.

## 1.4.19 ProcessEvent Type

**C++**

```
typedef void (* ProcessEvent)(float fPercent, float fDeviceBuffer, float fCache, double dBytesWritten, double dImageSize, void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
fPercent	Progress in % (0 – 100)
fDeviceBuffer	Device buffer usage in % (0 – 100)
fCache	Cache usage in % (0 – 100)
dBytesWritten	Bytes written
dImageSize	Bytes still to be written
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered during a deletion or burn, erase and finalize process. It shows the current progress of the process.

## 1.4.20 RAWDataEvent Type

**C++**

```
typedef int32 (* RAWDataEvent)(int32 nLBA, void* pBuffer, int32 nBufferSize, int32* nReaded, void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
nLBA	Current logical block address
pBuffer	Pointer to the data buffer
nBufferSize	Size of the buffer in bytes
nReaded	Pointer to retuned number of bytes readed
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered if the IsoSDK needs more data for burning the RAW Image file. This event will only occur if no file was added to a BS\_BRUNER\_RAW project.

## 1.4.21 RemoveFileEvent Type

**C++**

```
typedef void (* RemoveFileEvent)(const TCHAR* pcFullPath, const TCHAR* pcDestinationPath,
                                const TCHAR* pcFile, void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
pcFullPath	Path of the removed file (includes file name)
pcDestinationPath	Destination path of the file in the project
pcFile	Source path of the removed file (includes file name)
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when a file was removed from the project.

## 1.4.22 StartVerifyEvent Type

**C++**

```
typedef void (* StartVerifyEvent)(void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when the verify process will start.

## 1.4.23 uint16 Type

**C++**

```
typedef unsigned short uint16;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type uint16.

---

## 1.4.24 uint32 Type

**C++**

```
typedef unsigned int uint32;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type uint32.

---

## 1.4.25 uint8 Type

**C++**

```
typedef unsigned char uint8;
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is type uint8.

---

## 1.4.26 VerifyDoneEvent Type

**C++**

```
typedef void (* VerifyDoneEvent)(int32 nNumErrors, void* pUserData);
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Parameters**

Parameters	Description
nNumErrors	Counted errors during the verify process
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when the verify process is finished.

## 1.4.27 VerifyErrorEvent Type

### C++

```
typedef void (* VerifyErrorEvent)(const TCHAR* pcFileName, const TCHAR* pcError, void* pUserData);
```

### File

IsoSDKDefinitions.h (see page 517)

### Parameters

Parameters	Description
pcFileName	File name and path of the current file in process.
pcError	String description of the error.
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

### Description

This event is triggered if an error occurred during the current verify process.

## 1.4.28 VerifyFileEvent Type

### C++

```
typedef void (* VerifyFileEvent)(const TCHAR* pcFileName, void* pUserData);
```

### File

IsoSDKDefinitions.h (see page 517)

### Parameters

Parameters	Description
pcFileName	File name and path of the current file in process
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

### Description

This event will be triggered every time a new file will be verified.

## 1.4.29 VerifySectorEvent Type

### C++

```
typedef void (* VerifySectorEvent)(double nSector, double tSector, bool nSuccess, void* pUserData);
```

### File

IsoSDKDefinitions.h (see page 517)

### Parameters

Parameters	Description
nSector	Number of the current sector

tSector	Number of all sectors
nSuccess	Current sector valid or invalid
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered every time a new file will be verified.

---

## 1.4.30 VideoScanDoneEvent Type

**C++**

```
typedef void (* VideoScanDoneEvent)(const TCHAR* pcFileName, const TCHAR* pcError, int32 nErrorCode, struct SVideoFormat CurrentFormat, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
pcFileName	File name and path of the video file that was scanned
pcError	Decoding error. If no error occurred, an empty string or a null pointer will be returned
nErrorCode	Error ID when an error occurred. See Language file
CurrentFormat	A structure SVideoFormat (see page 105)
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered when the video scanning process is completed.

---

## 1.4.31 VideoScannerEvent Type

**C++**

```
typedef void (* VideoScannerEvent)(float fPercent, const TCHAR* pcFileName, void* pUserData);
```

**File**

IsoSDKDefinitions.h (see page 517)

**Parameters**

Parameters	Description
fPercent	Scanning progress in % (0 – 100)
pcFileName	File name and path of the video file that is being scanned
pUserData	This value is a custom defined pointer (e.g. the pointer to the window)

**Description**

This event will be triggered during a running scanning process. It shows the current progress of the process.

# 1.5 Macros

The following table lists macros in this documentation.

## Macros

Name	Description
<code>_stat64</code> (see page 305)	This is macro <code>_stat64</code> .
<code>_BS_LIBRARY_H</code> (see page 305)	This is macro <code>_BS_LIBRARY_H</code> .
<code>_ISOSDK_DEFINITIONS_H</code> (see page 306)	This is macro <code>_ISOSDK_DEFINITIONS_H</code> .
<code>_ISOSDK_EXPORT_H</code> (see page 306)	This is macro <code>_ISOSDK_EXPORT_H</code> .
<code>_max</code> (see page 306)	This is macro <code>_max</code> .
<code>_min</code> (see page 306)	This is macro <code>_min</code> .
<code>_FindData_t</code> (see page 307)	This is macro <code>_FindData_t</code> .
<code>_FindFirst</code> (see page 307)	This is macro <code>_FindFirst</code> .
<code>_MAX_PATH</code> (see page 307)	This is macro <code>_MAX_PATH</code> .
<code>BOOL2bool</code> (see page 307)	This is macro <code>BOOL2bool</code> .
<code>BS_ANALOG_AUDIO_PLAYBACK</code> (see page 308)	Drive feature: Device support playback analog audio.
<code>BS_API</code> (see page 308)	VideoCD / SuperVideoCD - MPEG file has a 16:9 display
<code>BS_AR_16TO9_DISPLAY</code> (see page 308)	Video CD / Super Video CD - MPEG file has a 16:9 display.
<code>BS_AR_221TO2_DISPLAY</code> (see page 308)	Video CD / Super Video CD - MPEG file has a 2.21:1 display.
<code>BS_AR_4TO3_DISPLAY</code> (see page 309)	Video CD / Super Video CD - The MPEG file has a 4:3 display.

	BS_AR_SQUARE_PIXELS (see page 309)	Video CD / Super Video CD - MPEG file has square pixels aspect ratio.
	BS_AR_UNKNOWN (see page 309)	Video CD / Super Video CD - MPEG file has an unknown aspect ratio.
	BS_ASPI_FROGASPI (see page 309)	The FrogAspi DLL. This is a very good replacement of the Adaptec interface. It supports all new interfaces.
	BS_ASPI_INTERNAL (see page 310)	Use the IsoSDK internal ASPI. This is recommended for all Linux and macOS versions and Windows from Windows 2000 and above.
	BS_ASPI_WNASPI (see page 310)	The default external ASPI layer like the one from Adaptec. Please note that this ASPI does neither support USB nor FireWire.
	BS_AUDIO_MP3 (see page 310)	Audio format is MP3 (MPEG Audio Layer 3).
	BS_AUDIO_NO (see page 310)	Audio format is unkown.
	BS_AUDIO_OGG (see page 311)	Audio format is Ogg Vorbis.
	BS_AUDIO_PCM (see page 311)	Audio format is PCM.
	BS_BARCODE_READ (see page 311)	Drive feature: Device is able to read the disc barcode information.
	BS_BLURAY_R (see page 311)	A disc identifier for disc type: Blu-ray BD-R
	BS_BLURAY_R_RRM (see page 312)	A disc identifier for disc type: Blu-ray BD-RRM
	BS_BLURAY_RE (see page 312)	A disc identifier for disc type: Blu-ray BD-RE

	BS_BLURAY_ROM ( <a href="#">see page 312</a> )	A disc identifier for disc type: Blu-ray BD-ROM
	BS_BOOL ( <a href="#">see page 312</a> )	This is macro BS_BOOL.
	BS_BOOL2bool ( <a href="#">see page 313</a> )	Audio format is MP3 (MPEG Audio Layer 3).
	BS_BT_AVERAGE ( <a href="#">see page 313</a> )	Audio encoding: Average Bitrate (CBR)
	BS_BT_CONSTANT ( <a href="#">see page 313</a> )	Audio encoding: Constant Bitrate (CBR)
	BS_BT_VARIABLE ( <a href="#">see page 313</a> )	Audio encoding: Variable Bitrate (VBR)
	BS_BURN_DEVICE ( <a href="#">see page 314</a> )	IsoSDK device type: Burn ( <a href="#">see page 121</a> ) Device.
	BS_BURNER_AUDIO ( <a href="#">see page 314</a> )	IsoSDK project type AudioCD.
	BS_BURNER_BLURAY ( <a href="#">see page 314</a> )	IsoSDK project type Blu-Ray disc. This project type is for Blu-ray Video. It do not contain a ISO bridge. For ISO/UDF Blu-ray please use BS_BURNER_MIXEDMODE ( <a href="#">see page 315</a> ) project type.
	BS_BURNER_CUE ( <a href="#">see page 314</a> )	IsoSDK project type Bin/Cue Image.
	BS_BURNER_DATA ( <a href="#">see page 315</a> )	IsoSDK project type Data CD/DVD. ISO9660 file system + extension.
	BS_BURNER_ISOUDF ( <a href="#">see page 315</a> )	IsoSDK project type Data CD/DVD with ISO UDF Bridge.
	BS_BURNER_MIXEDMODE ( <a href="#">see page 315</a> )	IsoSDK project type Mixed Mode CD (Audio/Data).

	BS_BURNER_RAW (see page 315)	IsoSDK project type RAW Image (Universal). This project type can burn any type of disc images until you know the details.
	BS_BURNER_SVCD (see page 316)	IsoSDK project type SuperVideoCD.
	BS_BURNER_UDFDVD (see page 316)	IsoSDK project type Data CD/DVD with UDF only.
	BS_BURNER_VCD (see page 316)	IsoSDK project type VideoCD.
	BS_BURNER_VIDEODVD (see page 316)	IsoSDK type VideoDVD (ISO/UDF 1.02). This project type contain folders for VideoDVD and special sorting features for VideoDVD.
	BS_C2_POINTERS (see page 317)	Drive feature: Device support reading C2 error pointers and C2 block error flags.
	BS_CALL (see page 317)	This is macro BS_CALL.
	BS_CD_R (see page 317)	A disc identifier for disc type: CD-R
	BS_CD_ROM (see page 317)	A disc identifier for disc type: CD-ROM
	BS_CD_RW (see page 318)	A disc identifier for disc type: CD-ReWritable
	BS_CD_TEXT_READ (see page 318)	Drive feature: Device is able to read the CD-TEXT information of a disc (subchannel).
	BS_CD_TEXT_WRITE (see page 318)	Drive feature: Device is able to write the CD-TEXT information of a disc (subchannel).
	BS_CDDA_COMMANDS (see page 318)	Drive feature: Device is able to work with CDDA command set.

	BS_CDDA_STREAM_IS_ACCURATE (see page 319)	Drive feature: The Logical Unit shall support an audio location without losing place to continue the READ CD command.
	BS_CDTCI_ARRANGER (see page 319)	CD-Text content items: Arranger field.
	BS_CDTCI_COMPOSER (see page 319)	CD-Text content items: Composer field.
	BS_CDTCI_MESSAGE (see page 319)	CD-Text content items: Message field.
	BS_CDTCI_PERFORMER (see page 320)	CD-Text content items: Performer field.
	BS_CDTCI_SONG_WRITER (see page 320)	CD-Text content items: Songwriter field.
	BS_CDTCI_TITLE (see page 320)	CD-Text content items: Title field.
	BS_CHANGER_SIDE_CHANGE_CAPABLE (see page 320)	Drive feature: The Logical Unit is capable of selecting both sides of discs.
	BS_CHANGER_SOFTWARE_SLOT_SELECTION (see page 321)	Drive feature: This bit controls the behavior of the LOAD/UNLOAD MEDIUM command when trying to load a Slot with no disc present.
	BS_CHANGER_SUPPORTS_DISC_PRESENT (see page 321)	Drive feature: The Logical Unit contains an embedded changer, and after a reset condition or a magazine change, the Logical Unit is capable of reporting the exact contents of the slots.

	BS_COMPOSITE_AUDIO_AND_VIDEO (see page 321)	Drive feature: The Logical Unit shall be capable of delivering a composite Audio and Video data stream from an independent digital port.
	BS_COMPRESSED_SIGNATURE (see page 321)	This is macro BS_COMPRESSED_SIGNATURE .
	BS_COMPRESSED_SIGNATURE_FOR_DRIVER (see page 322)	This is macro BS_COMPRESSED_SIGNATURE_FOR_DRIVER .
	BS_CONTINUE_LAST_SESSION (see page 322)	Default value to import last session of the inserted disc.
	BS_CONTINUE_NO_SESSION (see page 322)	Default value to do not import any available session.
	BS_CPRMAUTH (see page 322)	Device is able to CPRM authentication and key management.
	BS_CURRENT_DEVICE (see page 323)	Default device index of the current selected drive.
	BS_DAO_16 (see page 323)	Drive feature: Device can write DAO16 method.
	BS_DAO_96_PACK (see page 323)	Drive feature: Device can write DAO96 packet write method.
	BS_DAO_96_RAW (see page 323)	Drive feature: Device can write DAO96 raw write method.
	BS_DAO_RAW (see page 324)	Drive feature: Device can write DAO raw write method.
	BS_DD_CD_R (see page 324)	A disc identifier for disc type: Double Density Compact Disc - Recording
	BS_DD_CD_ROM (see page 324)	A disc identifier for disc type: Double Density Compact Disc - ROM

	BS_DDCD_RW (see page 324)	A disc identifier for disc type: Double Density Compact Disc - ReWritable
	BS_DEFAULT_LANGUAGE_FILE (see page 325)	The defined default language file name: "language.ini".
	BS_DEFAULT_REGISTRY_KEY (see page 325)	The defined default registry key: "HKEY_CURRENT_USER".
	BS_DEFAULT_REGISTRY_PATH (see page 325)	The default defined registry path: "\Software\IsoSDK\".
	BS_DEFECTMANAGEMENT (see page 325)	Drive feature: Device has defect management feature. Available to provide defect-free address space. For example BD recorders provide this feature.
	BS_DIGITAL_PORT_1 (see page 326)	Drive feature: The Logical Unit shall support digital output (IEC958) on port 1.
	BS_DIGITAL_PORT_2 (see page 326)	Drive feature: The Logical Unit shall support digital output (IEC958) on port 2.
	BS_DONT_SAVE_PATH (see page 326)	The file is created in the chDestinationPath of the project.
	BS_DVD_MRDL (see page 326)	A disc identifier for disc type: DVD-R Double Layer
	BS_DVD_PLUSR (see page 327)	A disc identifier for disc type: DVD+R
	BS_DVD_PLUSRW (see page 327)	A disc identifier for disc type: DVD+R ReWritable

	BS_DVD_R ( <a href="#">see page 327</a> )	A disc identifier for disc type: DVD-R
	BS_DVD_RAM ( <a href="#">see page 327</a> )	A disc identifier for disc type: DVD-RAM
	BS_DVD_RDL_PLUS ( <a href="#">see page 328</a> )	A disc identifier for disc type: DVD+R Doube Layer
	BS_DVD_ROM ( <a href="#">see page 328</a> )	A disc identifier for disc type: DVD-ROM
	BS_DVD_RW ( <a href="#">see page 328</a> )	A disc identifier for disc type: DVD-R ReWriteable
	BS_DVD_RW_RO ( <a href="#">see page 328</a> )	A disc identifier for disc type: DVD-R ReWriteable - Restricted Overwrite
	BS_DVD_RW_SR ( <a href="#">see page 329</a> )	A disc identifier for disc type: DVD-R ReWriteable - Sequential Recording
	BS_DVD_RWDL_PLUS ( <a href="#">see page 329</a> )	A disc identifier for disc type: DVD+R ReWriteable Double Layer
	BS_EJECT_INDIVIDUAL_OR_MAGAZINE ( <a href="#">see page 329</a> )	Drive feature: The Logical Unit shall support media eject via the START STOP UNIT command with the LoEj bit set.
	BS_EMT_BD ( <a href="#">see page 329</a> )	Medium type: Blu-Ray medium
	BS_EMT_CD_AUDIO ( <a href="#">see page 330</a> )	Medium type: CD-DA format
	BS_EMT_CD_ENHANCED ( <a href="#">see page 330</a> )	Medium type: CD-Enhanced format
	BS_EMT_CD_MIXED_MODE ( <a href="#">see page 330</a> )	Medium type: Mixed Mode CD format
	BS_EMT_CD_MULTISESSION ( <a href="#">see page 330</a> )	Medium type: Multi-session data CD
	BS_EMT_CD_ROM ( <a href="#">see page 331</a> )	Medium type: Data CD or CD of unknown format

	BS_EMT_CD_ROM_XA (see page 331)	Medium type: CD-eXtended Architecture format
	BS_EMT_DVD (see page 331)	Medium type: DVD medium
	BS_EMT_HDDVD (see page 331)	Medium type: HDDVD medium
	BS_ET_AAC (see page 332)	Encoding type: AAC ( <a href="#">Advanced Audio Codec</a> )
	BS_ET_FLAC (see page 332)	Encoding type: FLAC ( <a href="#">Free Lossless Audio Codec</a> ).
	BS_ET_MP3 (see page 332)	Encoding type: MP3 ( <a href="#">MPEG Audio Layer 3</a> ).
	BS_ET_MP4 (see page 332)	Encoding type: MEPG-4 ( <a href="#">Mp4</a> )
	BS_ET_OGG (see page 333)	Encoding type: <a href="#">Ogg Vorbis</a>
	BS_ET_OPUS (see page 333)	Encoding type: <a href="#">Opus</a>
	BS_ET_WMA (see page 333)	Encoding type: WMA (Windows Media Audio)
	BS_FA_ADVANCED_HIDDEN (see page 333)	File attribute: Advanced hidden files. Not valid in all OS. Use in combination with ISO files.
	BS_FA_ALL (see page 334)	File attribute: All attributes set.
	BS_FA_ARCHIVE (see page 334)	File attribute: Archive file.
	BS_FA_DIRECTORY (see page 334)	File attribute: Folder / Directory.
	BS_FA_HIDDEN (see page 334)	File attribute: Hidden file.
	BS_FA_READONLY (see page 335)	File attribute: Write protected files.
	BS_FA_SYSTEM (see page 335)	File attribute: System files.
	BS_FALSE (see page 335)	IsoSDK BOOL value: FALSE
	BS_FF_BIN (see page 335)	File format is BIN (2352).
	BS_FF_ISO (see page 336)	File format is ISO (2048)
	BS_FF_MPEG (see page 336)	File format is MPEG.
	BS_FF_WAVE (see page 336)	File format is WAVE (PCM).

	BS_FS_BOOTABLE (see page 336)	Disc file system: El Torito bootable extension to ISO 9660.
	BS_FS_ISO9660 (see page 337)	Disc file system: ISO 9660.
	BS_FS_JOLIET (see page 337)	Disc file system: Joliet extension to ISO 9660.
	BS_FS_ROCKRIDGE (see page 337)	Disc file system: Rock Ridge.
	BS_FS_UDF (see page 337)	Disc file system: Universal Disk Format.
	BS_FS_UNKNOWN (see page 338)	Disc file system: No file system or unknown file system.
	BS_GUI_ERROR_01 (see page 338)	GUI error string: Password and confirmation do not match.
	BS_GUI_ERROR_02 (see page 338)	GUI error string: Password is empty.
	BS_GUI_RESOURCE_01 (see page 338)	GUI resource string: CD/DVD Eraser (Title of erase dialog)
	BS_GUI_RESOURCE_02 (see page 339)	GUI resource string: CD/DVD Burner (Title of burn window).
	BS_GUI_RESOURCE_03 (see page 339)	GUI resource string: Device
	BS_GUI_RESOURCE_04 (see page 339)	GUI resource string: Progress
	BS_GUI_RESOURCE_05 (see page 339)	GUI resource string: Status
	BS_GUI_RESOURCE_06 (see page 340)	GUI resource string: Fast
	BS_GUI_RESOURCE_07 (see page 340)	GUI resource string: Complete
	BS_GUI_RESOURCE_08 (see page 340)	GUI resource string: Exit
	BS_GUI_RESOURCE_09 (see page 340)	GUI resource string: Settings
	BS_GUI_RESOURCE_10 (see page 341)	GUI resource string: Start
	BS_GUI_RESOURCE_11 (see page 341)	GUI resource string: Stop
	BS_GUI_RESOURCE_12 (see page 341)	GUI resource string: Simulate burning
	BS_GUI_RESOURCE_13 (see page 342)	GUI resource string: Finalize medium

	BS_GUI_RESOURCE_14 (see page 342)	GUI resource string: Finalize medium
	BS_GUI_RESOURCE_15 (see page 342)	GUI resource string: Speed
	BS_GUI_RESOURCE_16 (see page 342)	GUI resource string: Eject medium after burn
	BS_GUI_RESOURCE_17 (see page 343)	GUI resource string: Joliet file system
	BS_GUI_RESOURCE_18 (see page 343)	GUI resource string: Cache size
	BS_GUI_RESOURCE_19 (see page 343)	GUI resource string: 0.5 MB
	BS_GUI_RESOURCE_20 (see page 343)	GUI resource string: 64 MB
	BS_GUI_RESOURCE_21 (see page 344)	GUI resource string: Medium name
	BS_GUI_RESOURCE_22 (see page 344)	GUI resource string: Make boot medium
	BS_GUI_RESOURCE_23 (see page 344)	GUI resource string: No of Copies
	BS_GUI_RESOURCE_24 (see page 345)	GUI resource string: Medium
	BS_GUI_RESOURCE_25 (see page 345)	GUI resource string: Burning
	BS_GUI_RESOURCE_26 (see page 345)	GUI resource string: General
	BS_GUI_RESOURCE_27 (see page 345)	GUI resource string: Simulation was successful
	BS_GUI_RESOURCE_28 (see page 346)	GUI resource string: Burn (see page 121) the medium now?
	BS_GUI_RESOURCE_29 (see page 346)	GUI resource string: Verify data after burn
	BS_GUI_RESOURCE_30 (see page 346)	GUI resource string: DVD High-Compatibility-Mode
	BS_GUI_RESOURCE_31 (see page 346)	GUI resource string: Bytes Written:
	BS_GUI_RESOURCE_32 (see page 347)	GUI resource string: Elapsed Time:
	BS_GUI_RESOURCE_33 (see page 347)	GUI resource string: Buffer:

	BS_GUI_RESOURCE_34 (see page 347)	GUI resource string: Remaining Time:
	BS_GUI_RESOURCE_35 (see page 347)	GUI resource string: Other
	BS_GUI_RESOURCE_36 (see page 348)	GUI resource string: Compression
	BS_GUI_RESOURCE_37 (see page 348)	GUI resource string: Encryption
	BS_GUI_RESOURCE_38 (see page 348)	GUI resource string: Password
	BS_GUI_RESOURCE_39 (see page 349)	GUI resource string: UDF
	BS_GUI_RESOURCE_40 (see page 349)	GUI resource string: Version
	BS_GUI_RESOURCE_41 (see page 349)	GUI resource string: Partition type
	BS_GUI_RESOURCE_42 (see page 349)	GUI resource string: Save Log
	BS_GUI_RESOURCE_43 (see page 350)	GUI resource string: Select multiple devices
	BS_GUI_RESOURCE_44 (see page 350)	GUI resource string: Multiple devices
	BS_GUI_RESOURCE_45 (see page 350)	GUI resource string: Write method
	BS_GUI_RESOURCE_46 (see page 350)	GUI resource string: Confirm
	BS_GUI_RESOURCE_47 (see page 351)	GUI resource string: Write streams
	BS_GUI_RESOURCE_48 (see page 351)	GUI resource string: Rescan
	BS_GUI_RESOURCE_49 (see page 351)	GUI resource string: Auto Erase (see page 153)
	BS_GUI_RESOURCE_50 (see page 352)	GUI resource string: Rock Ridge File System
	BS_GUI_RESOURCE_51 (see page 352)	GUI resource string: Pad Data Tracks
	BS_GUI_RESOURCE_52 (see page 352)	GUI resource string: Eject (Erase (see page 153) Dialog)
	BS_GUI_RESOURCE_53 (see page 352)	GUI resource string: AVCHD disc
	BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLUMN (see page 353)	GUI resource string: Artist

	BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_COLUMN (see page 353)	GUI resource string: Category
	BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLUMN (see page 353)	GUI resource string: DiscID
	BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABEL (see page 354)	GUI resource string: Status
	BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE (see page 354)	GUI resource string: Title
	BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN (see page 354)	GUI resource string: Title
	BS_HD_DVD_R (see page 355)	A disc identifier for disc type: HDDVD R
	BS_HD_DVD_R_DL (see page 355)	A disc identifier for disc type: HDDVD R Double Layer
	BS_HD_DVD_RAM (see page 355)	A disc identifier for disc type: HDDVD-RAM
	BS_HD_DVD_ROM (see page 355)	A disc identifier for disc type: HDDVD-ROM
	BS_HD_DVD_RW (see page 356)	A disc identifier for disc type: HDDVD-ReWriteable
	BS_HD_DVD_RW_DL (see page 356)	A disc identifier for disc type: HDDVD ReWriteable Double Layer
	BS_IL_HIGH_DEBUG (see page 356)	Log info level: High. Use only for debugging.
	BS_IL_INFO (see page 356)	Log info level: Info. Use this for your own log files.
	BS_IL_LOW_DEBUG (see page 357)	Log info level: Low. Use only for debugging.
	BS_IL_MEDIUM_DEBUG (see page 357)	Log info level: Medium. Use only for debugging.
	BS_IIE_TOO_BIG_FILE (see page 357)	ISO level error: File is to Big.
	BS_IIE_TOO_LONG_DIRECTORY_NESTING (see page 357)	ISO level error: To long Directory
	BS_IMG_BIN (see page 358)	A image type BIN (RAW) with a blocksize of 2352.
	BS_IMG_ISO (see page 358)	A image type ISO with a blocksize of 2048.

	BS_IMGTASK_CREATE (see page 358)	The IsoSDK will create a image file.
	BS_IMGTASK_VERIFY (see page 358)	The IsoSDK will verify a image file.
	BS_IMPOPTS_COMMON (see page 359)	This is macro BS_IMPOPTS_C OMMON.
	BS_IMPOPTS_DECRYPT (see page 359)	This is macro BS_IMPOPTS_D ECRYPT.
	BS_IMPOPTS_EX (see page 359)	This is macro BS_IMPOPTS_E X.
	BS_IMPOPTS_UNCOMPRESS (see page 359)	This is macro BS_IMPOPTS_U NCOMPRESS.
	BS_INVALID_HANDLE (see page 360)	IsoSDK error code: The submitted handle is not valid or not initialized.
	BS_INVALID_TAG_HANDLE (see page 360)	IsoSDK error code: A created but not initialized Handle to receive Tags of a disc.
	BS_ISO_LEVEL_1 (see page 360)	Disc file system: ISO Level 1 Specification: <ol style="list-style-type: none"> <li>1. Name format is 8.3. Available chars: only capital letters and underscore.</li> <li>2. File extension length is limited to 3 chars.</li> <li>3. Directory can't has extension.</li> <li>4. File size limit is 2 GB.</li> </ol>

	BS_ISO_LEVEL_2 (see page 360)	Disc file system: ISO Level 2. Specification: 1. Max name length is 31 chars. 2. File size limit is 2 GB.
	BS_ISO_LEVEL_3 (see page 361)	Disc file system: ISO Level 3. Specification: 1. Max name length is 128 chars. 2. No file size limit.
	BS_ISO_LEVEL_ROMEO (see page 361)	Disc file system: ISO Level Romeo. Specification: 1. Max name length is 128 chars. 2. Lower case chars are allowed in file names.
	BS_ISRC_READ (see page 361)	Drive feature: Device is able to read the ISRC (International Standard Recording Code) info of a disc.
	BS_LABELFLASH (see page 362)	Drive feature: Device is able to write <a href="#">Labelflash</a> discs.
	BS_LAYER_JUMP_RECORDING (see page 362)	Drive feature: Device is able to perform Layer Jump Recording.
	BS_LIGHTSCRIBE (see page 362)	Drive feature: Device is able to write <a href="#">Lightscribe</a> discs.
	BS_LOCK_MEDIA (see page 362)	Drive feature: Device is able to lock the media.
	BS_LOCK_STATE (see page 363)	Drive feature: Device is able return the lock state.
	BS_LP_BOOL (see page 363)	This is macro BS_LP_BOOL.

	BS_LS_COMPLETE_SESSION ( <a href="#">see page 363</a> )	Disc session status: Complete.
	BS_LS_DAMAGED_SESSION ( <a href="#">see page 363</a> )	Disc session status: Session is damaged.
	BS_LS_EMPTY_SESSION ( <a href="#">see page 364</a> )	Disc session status: Session is empty.
	BS_LS_INCOMPLETE_SESSION ( <a href="#">see page 364</a> )	Disc session status: Session is incomplete.
	BS_MAX_SPEED ( <a href="#">see page 364</a> )	This represents the max available speed of the device.
	BS_METHOD_2_ADDRESSING_FIXED_PACKETS ( <a href="#">see page 364</a> )	Drive feature: Device support reading fixed packet tracks on CD-R/RW media where the Addressing type is Method 2.
	BS_MODE2_FORM1_READ ( <a href="#">see page 365</a> )	Drive feature: Device is able to read disc that are recorded with Mode2 Form1 format.
	BS_MODE2_FORM2_READ ( <a href="#">see page 365</a> )	Drive feature: Device is able to read disc that are recorded with Mode2 Form2 format.
	BS_MS_COMPLETE_DISK ( <a href="#">see page 365</a> )	Disc status: Disc is complete / finalized.
	BS_MS_EMPTY_DISK ( <a href="#">see page 365</a> )	Disc status: Disc is empty
	BS_MS_INCOMPLETE_DISK ( <a href="#">see page 366</a> )	Disc status: Disc is incomplete (not finalized).
	BS_MS_OTHER ( <a href="#">see page 366</a> )	Disc status: Unknown
	BS_MULTI_ERROR_01 ( <a href="#">see page 366</a> )	IsoSDK error code: The disc are not identical.
	BS_MULTI_ERROR_02 ( <a href="#">see page 366</a> )	IsoSDK error code: Invalid write speed.
	BS_MULTISESSION ( <a href="#">see page 367</a> )	Drive feature: Device is able to perform multi-session disc.
	BS_NDEF ( <a href="#">see page 367</a> )	This is macro BS_NDEF.

	BS_PACKET_WRITE (see page 367)	Drive feature: The drive can perform packet writing.
	BS_PARENTDIR_ONLY (see page 367)	Only the superordinate directory of the file will be created in chDestinationPath of the project. Example: chSourceDir = c:\tmp1\tmp2\test.tmp - only the subfolder tmp2 will be created.
	BS_PREVENT_JUMPER (see page 368)	Drive feature: The Logical Unit has a physical jumper named the Prevent/Allow Jumper and the jumper is present.
	BS_R_W_SUBCHANNELS_DEINT_AND_CORR (see page 368)	Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command with the returned data de-interleaved and error corrected
	BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ (see page 368)	Drive feature: The Logical Unit is capable of reading the raw R-W Sub-channel information from the Lead-in.
	BS_R_W_SUBCHANNELS_READ (see page 368)	Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command.
	BS_RDT_DATA (see page 369)	Input data contains UserData field

	BS_RDT_EDC_ECC (see page 369)	Input data contains ECC (Error Correction Code) /EDC (Error Detection Code) fields
	BS_RDT_SUBCH_PQ (see page 369)	Input data contains P-Q subchannel data in packed form
	BS_RDT_SUBCH_PW (see page 369)	Input data contains P-W subchannel data in raw form.
	BS_RDT_SUBCH_RW (see page 370)	Input data contains R-W subchannel data in raw form.
	BS_RDT_SUBHEADERS (see page 370)	Input data contains Subheaders field.
	BS_RDT_SYNC_HEADER (see page 370)	Input data contains Sync/Header field.
	BS_READ_BLURAY_R (see page 370)	Drive feature: Device is able to read Blu-Ray R.
	BS_READ_BLURAY_R_XL (see page 371)	Drive feature: Device is able to read BD-R XL.
	BS_READ_BLURAY_RE (see page 371)	Drive feature: Device is able to read Blu-Ray ReWritable.
	BS_READ_BLURAY_RE_XL (see page 371)	Drive feature: Device is able to read BD-RE XL (Re-Writable).
	BS_READ_BLURAY_ROM (see page 371)	Drive feature: Device is able to read Blu-Ray ROM.
	BS_READ_CDR (see page 372)	Drive feature: Device is able to read CD-R.
	BS_READ_CDRW (see page 372)	Drive feature: Device is able to read CD ReWritable.
	BS_READ_CDRW_CAV (see page 372)	Drive feature: Device is able to read CD-RW media that is designed for CAV recording.
	BS_READ_DEVICE (see page 372)	IsoSDK device type: Read Device

	BS_READ_DVD (see page 373)	Drive feature: Device is able to read DVD-R.
	BS_READ_DVD_DL (see page 373)	Drive feature: Device is able to read DVD-R Double Layer.
	BS_READ_DVD_MRDL (see page 373)	Drive feature: Device is able to read DVD-R Double Layer.
	BS_READ_DVD_RDL_PLUS (see page 373)	Drive feature: Device is able to read DVD+R Double Layer.
	BS_READ_DVD_RWDL_PLUS (see page 374)	Drive feature: Device is able to read DVD+ReWritable Double Layer.
	BS_READ_DVDR (see page 374)	Drive feature: Device is able to read DVD-R.
	BS_READ_DVDR_PLUS (see page 374)	Drive feature: Device is able to read DVD+R.
	BS_READ_DVDRAM (see page 374)	Drive feature: Device is able to read DVD-RAM.
	BS_READ_DVDRW (see page 375)	Drive feature: Device is able to read DVD-ReWritable.
	BS_READ_DVDRW_PLUS (see page 375)	Drive feature: Device is able to read DVD+ReWritable.
	BS_READ_HDDVD_R (see page 375)	Drive feature: Device is able to read HDDVD R.
	BS_READ_HDDVD_ROM (see page 375)	Drive feature: Device is able to read HDDVD Rom.
	BS_READ_HDDVD_RW (see page 376)	Drive feature: Device is able to read HDDVD ReWritable.
	BS_READ_MOUNT_RAINER (see page 376)	Drive feature: Device is able to read Mount Rainer.

	BS_RM_RAW ( <a href="#">see page 376</a> )	IsoSDK read mode: RAW (2352)
	BS_RM_RAW_SUBCHANNEL ( <a href="#">see page 376</a> )	IsoSDK read mode: RAW Subchannel (2352 +16)
	BS_RM_USERDATA ( <a href="#">see page 377</a> )	IsoSDK read mode: Userdata
	BS_RTF_AUDIO ( <a href="#">see page 377</a> )	Track type: Audio track
	BS_RTF_MODE1 ( <a href="#">see page 377</a> )	Track type: Mode 1 track
	BS_RTF_MODE2_FORM1 ( <a href="#">see page 377</a> )	Track type: <b>Mode 2 Form 1</b> track
	BS_RTF_MODE2_FORM2 ( <a href="#">see page 378</a> )	Track type: <b>Mode 2 Form 2</b> track
	BS_RTF_MODE2_FORMLESS ( <a href="#">see page 378</a> )	Track type: <b>Mode 2 formless</b> track
	BS_SCSI_ERROR_001 ( <a href="#">see page 378</a> )	IsoSDK error code: Operation in progress.
	BS_SCSI_ERROR_002 ( <a href="#">see page 378</a> )	IsoSDK error code: Cleaning requested.
	BS_SCSI_ERROR_004 ( <a href="#">see page 379</a> )	IsoSDK error code: No reference position found
	BS_SCSI_ERROR_005 ( <a href="#">see page 379</a> )	IsoSDK error code: Multiple peripheral devices selected.
	BS_SCSI_ERROR_006 ( <a href="#">see page 379</a> )	IsoSDK error code: Unreachable copy target.
	BS_SCSI_ERROR_007 ( <a href="#">see page 379</a> )	IsoSDK error code: Track following error.
	BS_SCSI_ERROR_008 ( <a href="#">see page 380</a> )	IsoSDK error code: Tracking servo failure.
	BS_SCSI_ERROR_009 ( <a href="#">see page 380</a> )	IsoSDK error code: Focus servo failure
	BS_SCSI_ERROR_01 ( <a href="#">see page 380</a> )	IsoSDK error code: No additional sense information
	BS_SCSI_ERROR_010 ( <a href="#">see page 380</a> )	IsoSDK error code: Spindle servo failure.
	BS_SCSI_ERROR_011 ( <a href="#">see page 381</a> )	IsoSDK error code: Head select fault.

	BS_SCSI_ERROR_012 (see page 381)	IsoSDK error code: Head select fault.
	BS_SCSI_ERROR_013 (see page 381)	IsoSDK error code: I-ec uncorrectable error.
	BS_SCSI_ERROR_014 (see page 381)	IsoSDK error code: Read error - Loss of streaming.
	BS_SCSI_ERROR_015 (see page 382)	IsoSDK error code: Recorded entity not found.
	BS_SCSI_ERROR_016 (see page 382)	IsoSDK error code: Record not found.
	BS_SCSI_ERROR_017 (see page 382)	IsoSDK error code: Random positioning error.
	BS_SCSI_ERROR_018 (see page 382)	IsoSDK error code: Parameter list length error.
	BS_SCSI_ERROR_019 (see page 383)	IsoSDK error code: Synchronous data transfer error.
	BS_SCSI_ERROR_02 (see page 383)	IsoSDK error code: I/O process terminated.
	BS_SCSI_ERROR_020 (see page 383)	IsoSDK error code: Miss compare during verify operation.
	BS_SCSI_ERROR_021 (see page 383)	IsoSDK error code: Invalid command operation code.
	BS_SCSI_ERROR_022 (see page 384)	IsoSDK error code: Logical block address out of range.
	BS_SCSI_ERROR_023 (see page 384)	IsoSDK error code: Invalid element address.
	BS_SCSI_ERROR_024 (see page 384)	IsoSDK error code: Invalid address for write.
	BS_SCSI_ERROR_025 (see page 384)	IsoSDK error code: Invalid release of persistent reservation.
	BS_SCSI_ERROR_026 (see page 385)	IsoSDK error code: Data decryption error.

	BS_SCSI_ERROR_027 (see page 385)	IsoSDK error code: Inline data length exceeded.
	BS_SCSI_ERROR_028 (see page 385)	IsoSDK error code: Invalid operation for copy source or destination.
	BS_SCSI_ERROR_029 (see page 385)	IsoSDK error code: Not ready to ready change, medium may have changed.
	BS_SCSI_ERROR_030 (see page 386)	IsoSDK error code: Import or export element accessed.
	BS_SCSI_ERROR_031 (see page 386)	IsoSDK error code: Power on, reset, or bus device reset occurred.
	BS_SCSI_ERROR_032 (see page 386)	IsoSDK error code: Power on occurred.
	BS_SCSI_ERROR_033 (see page 386)	IsoSDK error code: SCSI bus reset occurred.
	BS_SCSI_ERROR_034 (see page 387)	IsoSDK error code: Bus device reset function occurred.
	BS_SCSI_ERROR_035 (see page 387)	IsoSDK error code: Device internal reset.
	BS_SCSI_ERROR_036 (see page 387)	IsoSDK error code: Transceiver mode changed to single-ended.
	BS_SCSI_ERROR_037 (see page 387)	IsoSDK error code: Transceiver mode changed to LVD.
	BS_SCSI_ERROR_038 (see page 388)	IsoSDK error code: Reservations pre-empted.
	BS_SCSI_ERROR_039 (see page 388)	IsoSDK error code: Reservations released.
	BS_SCSI_ERROR_040 (see page 388)	IsoSDK error code: Registrations pre-empted.

	BS_SCSI_ERROR_041 (see page 388)	IsoSDK error code: Copy cannot execute since host cannot disconnect.
	BS_SCSI_ERROR_042 (see page 389)	IsoSDK error code: Current program area is not empty.
	BS_SCSI_ERROR_043 (see page 389)	IsoSDK error code: Current program area is empty.
	BS_SCSI_ERROR_044 (see page 389)	IsoSDK error code: Persistent prevent conflict.
	BS_SCSI_ERROR_045 (see page 389)	IsoSDK error code: Insufficient time for operation.
	BS_SCSI_ERROR_046 (see page 390)	IsoSDK error code: Zoned formatting failed due to spare linking.
	BS_SCSI_ERROR_047 (see page 390)	IsoSDK error code: Enclosure failure.
	BS_SCSI_ERROR_048 (see page 390)	IsoSDK error code: Enclosure services failure.
	BS_SCSI_ERROR_049 (see page 390)	IsoSDK error code: Unsupported enclosure function.
	BS_SCSI_ERROR_050 (see page 391)	IsoSDK error code: Enclosure services unavailable.
	BS_SCSI_ERROR_051 (see page 391)	IsoSDK error code: Enclosure services transfer failure.
	BS_SCSI_ERROR_052 (see page 391)	IsoSDK error code: Enclosure services transfer refused.
	BS_SCSI_ERROR_053 (see page 391)	IsoSDK error code: Invalid bits in identify message.
	BS_SCSI_ERROR_054 (see page 392)	IsoSDK error code: Target operating conditions have changed.

	BS_SCSI_ERROR_055 (see page 392)	IsoSDK error code: Microcode has been changed.
	BS_SCSI_ERROR_056 (see page 392)	IsoSDK error code: Changed operating definition.
	BS_SCSI_ERROR_057 (see page 392)	IsoSDK error code: Inquiry data has changed.
	BS_SCSI_ERROR_058 (see page 393)	IsoSDK error code: Component device attached.
	BS_SCSI_ERROR_059 (see page 393)	IsoSDK error code: Device identifier change.
	BS_SCSI_ERROR_060 (see page 393)	IsoSDK error code: Redundancy group created or modified.
	BS_SCSI_ERROR_061 (see page 393)	IsoSDK error code: Redundancy group deleted.
	BS_SCSI_ERROR_062 (see page 394)	IsoSDK error code: Spare created or modified.
	BS_SCSI_ERROR_063 (see page 394)	IsoSDK error code: Spare deleted.
	BS_SCSI_ERROR_064 (see page 394)	IsoSDK error code: Reported luns data has changed.
	BS_SCSI_ERROR_065 (see page 394)	IsoSDK error code: Echo buffer overwritten.
	BS_SCSI_ERROR_066 (see page 395)	IsoSDK error code: Message error.
	BS_SCSI_ERROR_067 (see page 395)	IsoSDK error code: Internal target failure.
	BS_SCSI_ERROR_068 (see page 395)	IsoSDK error code: Select or reselect failure.
	BS_SCSI_ERROR_069 (see page 395)	IsoSDK error code: Unsuccessful soft reset.
	BS_SCSI_ERROR_070 (see page 396)	IsoSDK error code: SCSI parity error.

	BS_SCSI_ERROR_071 (see page 396)	IsoSDK error code: Data phase crc error detected.
	BS_SCSI_ERROR_072 (see page 396)	IsoSDK error code: SCSI parity error detected during st data phase.
	BS_SCSI_ERROR_073 (see page 396)	IsoSDK error code: Information unit crc error detected.
	BS_SCSI_ERROR_074 (see page 397)	IsoSDK error code: Asynchronous information protection error detected.
	BS_SCSI_ERROR_075 (see page 397)	IsoSDK error code: Initiator detected error message received.
	BS_SCSI_ERROR_076 (see page 397)	IsoSDK error code: Invalid message error.
	BS_SCSI_ERROR_077 (see page 397)	IsoSDK error code: Data phase error.
	BS_SCSI_ERROR_078 (see page 398)	IsoSDK error code: Insufficient reservation resources.
	BS_SCSI_ERROR_079 (see page 398)	IsoSDK error code: Insufficient resources.
	BS_SCSI_ERROR_080 (see page 398)	IsoSDK error code: Insufficient registration resources.
	BS_SCSI_ERROR_081 (see page 398)	IsoSDK error code: Unable to recover table-of-contents.
	BS_SCSI_ERROR_082 (see page 399)	IsoSDK error code: Operator request or state change input.
	BS_SCSI_ERROR_083 (see page 399)	IsoSDK error code: Operator medium removal request.
	BS_SCSI_ERROR_084 (see page 399)	IsoSDK error code: Operator selected write protect.

	BS_SCSI_ERROR_085 (see page 399)	IsoSDK error code: Operator selected write permit.
	BS_SCSI_ERROR_086 (see page 400)	IsoSDK error code: Threshold condition met.
	BS_SCSI_ERROR_087 (see page 400)	IsoSDK error code: Failure prediction threshold exceeded.
	BS_SCSI_ERROR_088 (see page 400)	IsoSDK error code: Spare area exhaustion prediction threshold exceeded.
	BS_SCSI_ERROR_089 (see page 400)	IsoSDK error code: Failure prediction threshold exceeded (false).
	BS_SCSI_ERROR_090 (see page 401)	IsoSDK error code: Low power condition on.
	BS_SCSI_ERROR_091 (see page 401)	IsoSDK error code: Idle condition activated by timer.
	BS_SCSI_ERROR_092 (see page 401)	IsoSDK error code: Standby condition activated by timer.
	BS_SCSI_ERROR_093 (see page 401)	IsoSDK error code: Idle condition activated by command.
	BS_SCSI_ERROR_094 (see page 402)	IsoSDK error code: Standby condition activated by command.
	BS_SCSI_ERROR_095 (see page 402)	IsoSDK error code: End of user area encountered on this track.
	BS_SCSI_ERROR_096 (see page 402)	IsoSDK error code: Packet does not fit in available space.
	BS_SCSI_ERROR_097 (see page 402)	IsoSDK error code: Illegal mode for this track.

	BS_SCSI_ERROR_098 (see page 403)	IsoSDK error code: Invalid packet size.
	BS_SCSI_ERROR_099 (see page 403)	IsoSDK error code: Voltage fault.
	BS_SCSI_ERROR_100 (see page 403)	IsoSDK error code: Empty or partially written reserved track.
	BS_SCSI_ERROR_101 (see page 403)	IsoSDK error code: No more track reservations allowed.
	BS_SCSI_ERROR_102 (see page 404)	IsoSDK error code: CD control error.
	BS_SCSI_ERROR_103 (see page 404)	IsoSDK error code: Power calibration area almost full.
	BS_SCSI_ERROR_104 (see page 404)	IsoSDK error code: Power calibration area is full.
	BS_SCSI_ERROR_105 (see page 404)	IsoSDK error code: Power calibration area error.
	BS_SCSI_ERROR_106 (see page 405)	IsoSDK error code: RMA/PMA is almost full.
	BS_SCSI_ERROR_107 (see page 405)	IsoSDK error code: Failed to request sense.
	BS_SCSI_ERROR_108 (see page 405)	IsoSDK error code: Medium load or eject failed.
	BS_SCSI_ERROR_109 (see page 405)	IsoSDK error code: Medium removal prevented.
	BS_SCSI_ERROR_110 (see page 406)	IsoSDK error code: Invalid write crossing layer jump.
	BS_SCSI_ERROR_111 (see page 406)	IsoSDK error code: Illegal function.
	BS_SCSI_ERROR_112 (see page 406)	IsoSDK error code: Format-layer may have changed.
	BS_SCSI_ERROR_113 (see page 406)	IsoSDK error code: Enclosure services checksum error.

	BS_SCSI_ERROR_114 (see page 407)	IsoSDK error code: System resource faulture.
	BS_SCSI_ERROR_115 (see page 407)	IsoSDK error code: RMZ extention is not allowed.
	BS_SCSI_ERROR_116 (see page 407)	IsoSDK error code: No more test zone extensions are allowed.
	BS_SCSI_ERROR_117 (see page 407)	IsoSDK error code: Current power calibration area is almost full.
	BS_SCSI_ERROR_118 (see page 408)	IsoSDK error code: Current power calibration area is full.
	BS_SCSI_ERROR_119 (see page 408)	IsoSDK error code: RDZ is full.
	BS_SCSI_ERROR_ALLOC_01 (see page 408)	IsoSDK error code: Program memory area update failure.
	BS_SCSI_ERROR_ALLOC_02 (see page 408)	IsoSDK error code: Program memory area is full.
	BS_SCSI_ERROR_ASPI_01 (see page 409)	IsoSDK error code: An ASPI error occurred.
	BS_SCSI_ERROR_ASPI_02 (see page 409)	IsoSDK error code: Invalid host adapter.
	BS_SCSI_ERROR_ASPI_03 (see page 409)	IsoSDK error code: No device has been specified.
	BS_SCSI_ERROR_ASPI_04 (see page 409)	IsoSDK error code: ASPI is busy.
	BS_SCSI_ERROR_ASPI_05 (see page 410)	IsoSDK error code: Bus reset.
	BS_SCSI_ERROR_ASPI_06 (see page 410)	IsoSDK error code: Invalid device ID or device is not a CD/DVD or BD drive.
	BS_SCSI_ERROR_ASPI_07 (see page 410)	IsoSDK error code: The device is busy.
	BS_SCSI_ERROR_ASPI_08 (see page 410)	IsoSDK error code: The target is busy.

	BS_SCSI_ERROR_ASPI_09 (see page 411)	IsoSDK error code: Target reservation conflict.
	BS_SCSI_ERROR_ASPI_10 (see page 411)	IsoSDK error code: The target queue is full.
	BS_SCSI_ERROR_ATT_01 (see page 411)	IsoSDK error code: Warning.
	BS_SCSI_ERROR_ATT_02 (see page 411)	IsoSDK error code: Warning - Specified temperature exceeded.
	BS_SCSI_ERROR_ATT_03 (see page 412)	IsoSDK error code: Warning - Enclosure degraded.
	BS_SCSI_ERROR_ATT_04 (see page 412)	IsoSDK error code: Warning - background self-test failed.
	BS_SCSI_ERROR_ATT_05 (see page 412)	IsoSDK error code: Warning – background pre-scan detected medium error.
	BS_SCSI_ERROR_AUDIO_01 (see page 412)	IsoSDK error code: Audio play operation in progress.
	BS_SCSI_ERROR_AUDIO_02 (see page 413)	IsoSDK error code: Audio play operation paused.
	BS_SCSI_ERROR_AUDIO_03 (see page 413)	IsoSDK error code: Audio play operation successfully completed.
	BS_SCSI_ERROR_AUDIO_04 (see page 413)	IsoSDK error code: Audio play operation stopped due to error.
	BS_SCSI_ERROR_AUDIO_05 (see page 413)	IsoSDK error code: No current audio status to return.
	BS_SCSI_ERROR_CDB_01 (see page 414)	IsoSDK error code: Invalid field in cdb.
	BS_SCSI_ERROR_CDB_02 (see page 414)	IsoSDK error code: CDB decryption error.
	BS_SCSI_ERROR_CIRC_01 (see page 414)	IsoSDK error code: Circ unrecovered error.

	BS_SCSI_ERROR_COMMAND_02 (see page 414)	IsoSDK error code: Command sequence error.
	BS_SCSI_ERROR_COMMAND_03 (see page 415)	IsoSDK error code: Commands cleared by another initiator.
	BS_SCSI_ERROR_COMMAND_04 (see page 415)	IsoSDK error code: Command phase error.
	BS_SCSI_ERROR_COMMAND_05 (see page 415)	IsoSDK error code: Overlapped commands attempted.
	BS_SCSI_ERROR_CRC_01 (see page 415)	IsoSDK error code: De-compression CRC error.
	BS_SCSI_ERROR_DCSS_01 (see page 416)	IsoSDK error code: Copy protection key exchange failure - authentication failure.
	BS_SCSI_ERROR_DCSS_02 (see page 416)	IsoSDK error code: Copy protection key exchange failure - key not present.
	BS_SCSI_ERROR_DCSS_03 (see page 416)	IsoSDK error code: Copy protection key exchange failure - key not established.
	BS_SCSI_ERROR_DCSS_04 (see page 416)	IsoSDK error code: Read of scrambled sector without authentication.
	BS_SCSI_ERROR_DCSS_05 (see page 417)	IsoSDK error code: Medium region code is mismatched to logical unit region.
	BS_SCSI_ERROR_DCSS_06 (see page 417)	IsoSDK error code: Drive region must be permanent/region reset count error.
	BS_SCSI_ERROR_DCSS_07 (see page 417)	IsoSDK error code: Insufficient block count for binding nonce recording.

	BS_SCSI_ERROR_DCSS_08 (see page 417)	IsoSDK error code: Conflict in binding nonce recording.
	BS_SCSI_ERROR_DECOM_01 (see page 418)	IsoSDK error code: Cannot decompress using declared algorithm.
	BS_SCSI_ERROR_DISK_01 (see page 418)	IsoSDK error code: Incompatible medium installeds.
	BS_SCSI_ERROR_DISK_02 (see page 418)	IsoSDK error code: Cannot read medium - unknown format.
	BS_SCSI_ERROR_DISK_03 (see page 418)	IsoSDK error code: Cannot read medium - incompatible format.
	BS_SCSI_ERROR_DISK_04 (see page 419)	IsoSDK error code: Cleaning cartridge installed.
	BS_SCSI_ERROR_DISK_05 (see page 419)	IsoSDK error code: Cannot write medium - unknown format.
	BS_SCSI_ERROR_DISK_06 (see page 419)	IsoSDK error code: Cannot write medium - incompatible format.
	BS_SCSI_ERROR_DISK_07 (see page 419)	IsoSDK error code: Cannot format medium - incompatible medium.
	BS_SCSI_ERROR_DISK_08 (see page 420)	IsoSDK error code: Cleaning failure.
	BS_SCSI_ERROR_DISK_09 (see page 420)	IsoSDK error code: Cannot write - application code mismatch.
	BS_SCSI_ERROR_DISK_10 (see page 420)	IsoSDK error code: Current session not fixated for append.
	BS_SCSI_ERROR_DISK_11 (see page 420)	IsoSDK error code: Medium not formatted.
	BS_SCSI_ERROR_DISK_12 (see page 421)	IsoSDK error code: Medium format corrupted.

	BS_SCSI_ERROR_DISK_13 (see page 421)	IsoSDK error code: Format command failed.
	BS_SCSI_ERROR_DISK_14 (see page 421)	IsoSDK error code: Medium not present.
	BS_SCSI_ERROR_DISK_15 (see page 421)	IsoSDK error code: Medium not present - tray closed.
	BS_SCSI_ERROR_DISK_16 (see page 422)	IsoSDK error code: Medium not present - tray open.
	BS_SCSI_ERROR_DISK_17 (see page 422)	IsoSDK error code: Medium not present - loadable.
	BS_SCSI_ERROR_DISK_18 (see page 422)	IsoSDK error code: Medium not present - medium auxiliary memory accessible.
	BS_SCSI_ERROR_DISK_19 (see page 422)	IsoSDK error code: Medium destination element full.
	BS_SCSI_ERROR_DISK_20 (see page 423)	IsoSDK error code: Medium source element empty.
	BS_SCSI_ERROR_DISK_21 (see page 423)	IsoSDK error code: End of medium reached.
	BS_SCSI_ERROR_DISK_22 (see page 423)	IsoSDK error code: Medium magazine not accessible.
	BS_SCSI_ERROR_DISK_23 (see page 423)	IsoSDK error code: Medium magazine removed.
	BS_SCSI_ERROR_DISK_24 (see page 424)	IsoSDK error code: Medium magazine inserted.
	BS_SCSI_ERROR_DISK_25 (see page 424)	IsoSDK error code: Medium magazine locked.
	BS_SCSI_ERROR_DISK_26 (see page 424)	IsoSDK error code: Medium magazine unlocked.
	BS_SCSI_ERROR_DISK_27 (see page 424)	IsoSDK error code: Medium loadable.

	BS_SCSI_ERROR_DISK_28 (see page 425)	IsoSDK error code: Medium auxiliary memory accessible.
	BS_SCSI_ERROR_DISK_29 (see page 425)	IsoSDK error code: The disc is not writable.
	BS_SCSI_ERROR_DISK_30 (see page 425)	IsoSDK error code: Wrong medium.
	BS_SCSI_ERROR_DISK_31 (see page 425)	IsoSDK error code: No medium.
	BS_SCSI_ERROR_DISK_32 (see page 426)	IsoSDK error code: The disc is not erasable.
	BS_SCSI_ERROR_DISK_33 (see page 426)	IsoSDK error code: Medium error.
	BS_SCSI_ERROR_DISK_34 (see page 426)	IsoSDK error code: Erase (see page 153) disc before writing.
	BS_SCSI_ERROR_DISK_35 (see page 426)	IsoSDK error code: The disc is not writable with current write method.
	BS_SCSI_ERROR_DISK_36 (see page 427)	IsoSDK error code: Cannot write medium - unsupported medium version.
	BS_SCSI_ERROR_DISK_37 (see page 427)	IsoSDK error code: Medium is not writeable.
	BS_SCSI_ERROR_DRIVE_01 (see page 427)	IsoSDK error code: Error reading UPC / EAN number.
	BS_SCSI_ERROR_DRIVE_02 (see page 427)	IsoSDK error code: Error reading ISRC number
	BS_SCSI_ERROR_EXT_01 (see page 428)	IsoSDK error code: Error detected by third party temporary initiator.
	BS_SCSI_ERROR_EXT_02 (see page 428)	IsoSDK error code: Third party device failure.
	BS_SCSI_ERROR_LOG_01 (see page 428)	IsoSDK error code: Error log overflow.
	BS_SCSI_ERROR_LOG_02 (see page 428)	IsoSDK error code: Log exception.

	BS_SCSI_ERROR_LOG_03 (see page 429)	IsoSDK error code: Log counter at maximum.
	BS_SCSI_ERROR_LOG_04 (see page 429)	IsoSDK error code: Log list codes exhausted.
	BS_SCSI_ERROR_MECH_01 (see page 429)	IsoSDK error code: Mechanical positioning error.
	BS_SCSI_ERROR_MECH_02 (see page 429)	IsoSDK error code: Positioning error detected by read of medium.
	BS_SCSI_ERROR_MECH_03 (see page 430)	IsoSDK error code: Mechanical positioning or changer error.
	BS_SCSI_ERROR_PARAM_01 (see page 430)	IsoSDK error code: Invalid field in parameter list.
	BS_SCSI_ERROR_PARAM_02 (see page 430)	IsoSDK error code: Parameter not supported.
	BS_SCSI_ERROR_PARAM_03 (see page 430)	IsoSDK error code: Parameter value invalid.
	BS_SCSI_ERROR_PARAM_04 (see page 431)	IsoSDK error code: Threshold parameters not supported.
	BS_SCSI_ERROR_PARAM_05 (see page 431)	IsoSDK error code: Parameters changed.
	BS_SCSI_ERROR_PARAM_06 (see page 431)	IsoSDK error code: Mode parameters changed.
	BS_SCSI_ERROR_PARAM_07 (see page 431)	IsoSDK error code: Log parameters changed.
	BS_SCSI_ERROR_PARAM_08 (see page 432)	IsoSDK error code: Rounded parameter.
	BS_SCSI_ERROR_PARAM_09 (see page 432)	IsoSDK error code: Saving parameters not supported.
	BS_SCSI_ERROR_PARAM_10 (see page 432)	IsoSDK error code: A bad parameter has been passed in.

	BS_SCSI_ERROR_READ_01 (see page 432)	IsoSDK error code: Unrecovered read error.
	BS_SCSI_ERROR_READ_02 (see page 433)	IsoSDK error code: Read retries exhausted.
	BS_SCSI_ERROR_RECOVER_01 (see page 433)	IsoSDK error code: Recovered data with no error correction applied.
	BS_SCSI_ERROR_RECOVER_02 (see page 433)	IsoSDK error code: Recovered data with retries.
	BS_SCSI_ERROR_RECOVER_03 (see page 433)	IsoSDK error code: Recovered data with positive head offset.
	BS_SCSI_ERROR_RECOVER_04 (see page 434)	IsoSDK error code: Recovered data with negative head offset.
	BS_SCSI_ERROR_RECOVER_05 (see page 434)	IsoSDK error code: Recovered data with retries and/or CIRC applied.
	BS_SCSI_ERROR_RECOVER_06 (see page 434)	IsoSDK error code: Recovered data using previous sector id.
	BS_SCSI_ERROR_RECOVER_07 (see page 434)	IsoSDK error code: Recovered data without ecc - recommend reassignment.
	BS_SCSI_ERROR_RECOVER_08 (see page 435)	IsoSDK error code: Recovered data without ecc - recommend rewrite.
	BS_SCSI_ERROR_RECOVER_09 (see page 435)	IsoSDK error code: Recovered data without ecc - data rewritten.
	BS_SCSI_ERROR_RECOVER_10 (see page 435)	IsoSDK error code: Recovered data with error correction applied.
	BS_SCSI_ERROR_RECOVER_11 (see page 435)	IsoSDK error code: Recovered data with error corr. & retries applied.

	BS_SCSI_ERROR_RECOVER_12 (see page 436)	IsoSDK error code: Recovered data - data auto-reallocated.
	BS_SCSI_ERROR_RECOVER_13 (see page 436)	IsoSDK error code: Recovered data with CIRC.
	BS_SCSI_ERROR_RECOVER_14 (see page 436)	IsoSDK error code: Recovered data with I-ec.
	BS_SCSI_ERROR_RECOVER_15 (see page 436)	IsoSDK error code: Recovered data - recommend reassignment.
	BS_SCSI_ERROR_RECOVER_16 (see page 437)	IsoSDK error code: Recovered data - recommend rewrite.
	BS_SCSI_ERROR_RECOVER_17 (see page 437)	IsoSDK error code: Recovered data with linking.
	BS_SCSI_ERROR_SEGM_01 (see page 437)	IsoSDK error code: Too many segment descriptors.
	BS_SCSI_ERROR_SEGM_02 (see page 437)	IsoSDK error code: Unsupported segment descriptor type code.
	BS_SCSI_ERROR_SEGM_03 (see page 438)	IsoSDK error code: Unexpected inexact segment.
	BS_SCSI_ERROR_SEGM_04 (see page 438)	IsoSDK error code: Copy segment granularity violation.
	BS_SCSI_ERROR_SESSION_01 (see page 438)	IsoSDK error code: Session fixation error.
	BS_SCSI_ERROR_SESSION_02 (see page 438)	IsoSDK error code: Session fixation error writing lead-in.
	BS_SCSI_ERROR_SESSION_03 (see page 439)	IsoSDK error code: Session fixation error writing lead-out.
	BS_SCSI_ERROR_SESSION_04 (see page 439)	IsoSDK error code: Session fixation error - incomplete track in session.

	BS_SCSI_ERROR_TARGET_01 (see page 439)	IsoSDK error code: Copy target device not reachable.
	BS_SCSI_ERROR_TARGET_02 (see page 439)	IsoSDK error code: Incorrect copy target device type.
	BS_SCSI_ERROR_TARGET_03 (see page 440)	IsoSDK error code: Copy target device data underrun.
	BS_SCSI_ERROR_TARGET_04 (see page 440)	IsoSDK error code: Session fixation error - incomplete track in session.
	BS_SCSI_ERROR_UNIT_01 (see page 440)	IsoSDK error code: No seek complete.
	BS_SCSI_ERROR_UNIT_02 (see page 440)	IsoSDK error code: Logical unit not ready, cause not reportable.
	BS_SCSI_ERROR_UNIT_03 (see page 441)	IsoSDK error code: Logical unit is in process of becoming ready.
	BS_SCSI_ERROR_UNIT_04 (see page 441)	IsoSDK error code: Logical unit not ready, initializing command required.
	BS_SCSI_ERROR_UNIT_05 (see page 441)	IsoSDK error code: Logical unit not ready, manual intervention required.
	BS_SCSI_ERROR_UNIT_06 (see page 441)	IsoSDK error code: Logical unit not ready, format in progress.
	BS_SCSI_ERROR_UNIT_07 (see page 442)	IsoSDK error code: Logical unit not ready, operation in progress.
	BS_SCSI_ERROR_UNIT_08 (see page 442)	IsoSDK error code: Logical unit not ready, long write in progress.

	BS_SCSI_ERROR_UNIT_09 (see page 442)	IsoSDK error code: Logical unit not ready, self-test in progress.
	BS_SCSI_ERROR_UNIT_10 (see page 442)	IsoSDK error code: Logical unit does not respond to selection.
	BS_SCSI_ERROR_UNIT_11 (see page 443)	IsoSDK error code: Logical unit communication failure.
	BS_SCSI_ERROR_UNIT_12 (see page 443)	IsoSDK error code: Logical unit communication time-out.
	BS_SCSI_ERROR_UNIT_13 (see page 443)	IsoSDK error code: Logical unit communication parity error.
	BS_SCSI_ERROR_UNIT_14 (see page 443)	IsoSDK error code: Logical unit communication CRC error (Ultra-DMA/32).
	BS_SCSI_ERROR_UNIT_16 (see page 444)	IsoSDK error code: Logical unit not supported.
	BS_SCSI_ERROR_UNIT_17 (see page 444)	IsoSDK error code: Logical unit has not self-configured yet.
	BS_SCSI_ERROR_UNIT_18 (see page 444)	IsoSDK error code: Logical unit failure.
	BS_SCSI_ERROR_UNIT_19 (see page 444)	IsoSDK error code: Timeout on logical unit.
	BS_SCSI_ERROR_UNIT_20 (see page 445)	IsoSDK error code: Logical unit failed self-test.
	BS_SCSI_ERROR_UNIT_21 (see page 445)	IsoSDK error code: Logical unit unable to update self-test log.
	BS_SCSI_ERROR_UNIT_22 (see page 445)	IsoSDK error code: Logical unit failed self-configuration.

	BS_SCSI_ERROR_UNIT_23 (see page 445)	IsoSDK error code: Logical unit failure prediction threshold exceeded.
	BS_SCSI_ERROR_VOL_01 (see page 446)	IsoSDK error code: Volume set created or modified.
	BS_SCSI_ERROR_VOL_02 (see page 446)	IsoSDK error code: Volume set deleted.
	BS_SCSI_ERROR_VOL_03 (see page 446)	IsoSDK error code: Volume set de-assigned.
	BS_SCSI_ERROR_VOL_04 (see page 446)	IsoSDK error code: Volume set reassigned.
	BS_SCSI_ERROR_WRITE_01 (see page 447)	IsoSDK error code: General write error.
	BS_SCSI_ERROR_WRITE_02 (see page 447)	IsoSDK error code: Write error - Recovery needed.
	BS_SCSI_ERROR_WRITE_03 (see page 447)	IsoSDK error code: Write error - Recovery failed.
	BS_SCSI_ERROR_WRITE_04 (see page 447)	IsoSDK error code: Write error - Loss of streaming.
	BS_SCSI_ERROR_WRITE_05 (see page 448)	IsoSDK error code: Write error - Padding blocks added.
	BS_SCSI_ERROR_WRITE_06 (see page 448)	IsoSDK error code: Write protected.
	BS_SCSI_ERROR_WRITE_07 (see page 448)	IsoSDK error code: Hardware write protected.
	BS_SCSI_ERROR_WRITE_08 (see page 448)	IsoSDK error code: Associated write protect.
	BS_SCSI_ERROR_WRITE_09 (see page 449)	IsoSDK error code: Persistent write protect.
	BS_SCSI_ERROR_WRITE_10 (see page 449)	IsoSDK error code: Permanent write protect.
	BS_SCSI_ERROR_WRITE_11 (see page 449)	IsoSDK error code: Conditional write protect.

	BS_SCSI_ERROR_WRITE_12 (see page 449)	IsoSDK error code: A write-error occurred.
	BS_SDK_COMPENC_BOTH (see page 450)	Medium Information: disc or image is compressed and encrypted.
	BS_SDK_COMPENC_COMPRESSED (see page 450)	Medium Information: disc or image is compressed.
	BS_SDK_COMPENC_ENCRYPTED (see page 450)	Medium Information: disc or image is encrypted.
	BS_SDK_CUE_ERROR_COMMAND_01 (see page 450)	IsoSDK error code: Invalid command.
	BS_SDK_CUE_ERROR_COMMAND_06 (see page 451)	IsoSDK error code: Command failed.
	BS_SDK_CUE_ERROR_FIELD (see page 451)	IsoSDK error code: Invalid field.
	BS_SDK_CUE_ERROR_FILE (see page 451)	IsoSDK error code: File not found.
	BS_SDK_CUE_ERROR_SENDING_CUE (see page 451)	IsoSDK error code: Send cue sheet failed.
	BS_SDK_CUE_ERROR_UEOI (see page 452)	IsoSDK error code: Unexpected end of line.
	BS_SDK_ERROR_ABORTED (see page 452)	IsoSDK error code: Operation aborted by user.
	BS_SDK_ERROR_ABORTEDCOMMAND (see page 452)	IsoSDK error code: Interface reported aborted comand was thrown.
	BS_SDK_ERROR_BAD_CAPABILITY_NAME (see page 452)	Capabilities error: A not existing capability name was submitted.
	BS_SDK_ERROR_BAD_REQUEST (see page 453)	IsoSDK error code: Bad request.
	BS_SDK_ERROR_BIN_FILE_NOT_FOUND (see page 453)	IsoSDK error code: The bin file could not be found.
	BS_SDK_ERROR_BUFFERALIGNMENT (see page 453)	IsoSDK error code: Buffer alignment error.

	BS_SDK_ERROR_BUFFERTOOBIG (see page 453)	IsoSDK error code: The buffer is too big.
	BS_SDK_ERROR_BURN_IN_PROGRESS (see page 454)	IsoSDK error code: Not allowed while burning.
	BS_SDK_ERROR_CDTEXT_NOT_FOUND (see page 454)	IsoSDK error code: CDText is not available on the audio disc.
	BS_SDK_ERROR_CHECKCONDITION (see page 454)	IsoSDK error code: Check condition.
	BS_SDK_ERROR_COMPENC_GENERALERROR (see page 454)	IsoSDK error code: General error while read compress/encrypt identifier.
	BS_SDK_ERROR_COMPENC_READ10_IDENT (see page 455)	IsoSDK error code: No IsoSDK identifier found on disc or image.
	BS_SDK_ERROR_COMPRESSION_CONFLICT (see page 455)	IsoSDK error code: A compression conflict occurred.
	BS_SDK_ERROR_COMPRESSION_NOINDXTABLE (see page 455)	IsoSDK error code: No Indexes table found.
	BS_SDK_ERROR_COMPRESSION_NOSIGNATURE (see page 455)	IsoSDK error code: No IsoSDK compression signature found.
	BS_SDK_ERROR_COMPRESSION_READ10 (see page 456)	IsoSDK error code: Error while operate Read10 from image or disc. Maybe not data.
	BS_SDK_ERROR_COMPRESSION_READMEMF (see page 456)	IsoSDK error code: Error while read data from MemFile. Maybe empty.
	BS_SDK_ERROR_COMPRESSION_SEEKMEMFILE (see page 456)	IsoSDK error code: Error while seek inside MemFile. Maybe no data.
	BS_SDK_ERROR_COPYABORTED (see page 457)	IsoSDK error code: Copy aborted.

	BS_SDK_ERROR_CORRUPT_OR_INVALID_CUE_FILE ( <a href="#">see page 457</a> )	IsoSDK error code: The cue file is corrupt or completely invalid.
	BS_SDK_ERROR_CREATEFILE ( <a href="#">see page 457</a> )	IsoSDK error code: IsoSDK was not able to create the file.
	BS_SDK_ERROR_DATAOVERRUN ( <a href="#">see page 457</a> )	IsoSDK error code: A data-overrun occurred.
	BS_SDK_ERROR_DATAPROTECT ( <a href="#">see page 458</a> )	IsoSDK error code: Data protected.
	BS_SDK_ERROR_DATARECOVERERROR ( <a href="#">see page 458</a> )	IsoSDK error code: The IsoSDK was not able to recover the data.
	BS_SDK_ERROR_DEVICE_LOCKED ( <a href="#">see page 458</a> )	IsoSDK error code: The device is locked. Maybe by another software / process.
	BS_SDK_ERROR_EMPTY_PASSWORD ( <a href="#">see page 458</a> )	IsoSDK error code: Empty passwords are not allowed.
	BS_SDK_ERROR_ENCLIB_NOT_FOUND ( <a href="#">see page 459</a> )	IsoSDK error code: bass library was not found in directory.
	BS_SDK_ERROR_ENCRYPTION_CONFLICT ( <a href="#">see page 459</a> )	IsoSDK error code: Encryption conflict.
	BS_SDK_ERROR_ERASECHECK ( <a href="#">see page 459</a> )	IsoSDK error code: Erase ( <a href="#">see page 153</a> ) check.
	BS_SDK_ERROR_ERRINVALIDFILENAME ( <a href="#">see page 459</a> )	IsoSDK error code: Error importing session, file name length exceeds 120 chars.
	BS_SDK_ERROR_FILE_EXISTS ( <a href="#">see page 460</a> )	IsoSDK error code: The file already exists.
	BS_SDK_ERROR_FILE_OPEN ( <a href="#">see page 460</a> )	IsoSDK error code: File could not be opened.
	BS_SDK_ERROR_FILEINUSE ( <a href="#">see page 460</a> )	IsoSDK error code: File could not be opened.

	BS_SDK_ERROR_FILEMARK (see page 460)	IsoSDK error code: Filemark error.
	BS_SDK_ERROR_GENERAL (see page 461)	IsoSDK error code: A general error occurred.
	BS_SDK_ERROR_HARDWAREERROR (see page 461)	IsoSDK error code: Hardware error.
	BS_SDK_ERROR_ILLEGALLLENGTH (see page 461)	IsoSDK error code: Illegal length of file name or path.
	BS_SDK_ERROR_IMPORTSESSION (see page 461)	IsoSDK error code: Error importing session.
	BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE (see page 462)	IsoSDK error code: Selected project type is incompatible with last recorded session.
	BS_SDK_ERROR_INCORRECTLENGTH (see page 462)	IsoSDK error code: Incorrect length of the given file name or path.
	BS_SDK_ERROR_INVALID_DEST_PATH (see page 462)	IsoSDK error code: An invalid destination path has been passed in.
	BS_SDK_ERROR_INVALID_DIR_INDEX (see page 462)	IsoSDK error code: Invalid directory index.
	BS_SDK_ERROR_INVALID_FILE_FORMAT (see page 463)	IsoSDK error code: The file has an invalid format.
	BS_SDK_ERROR_INVALID_FILE_NAME (see page 463)	IsoSDK error code: An invalid file name has been passed in.
	BS_SDK_ERROR_INVALID_HANDLE (see page 463)	IsoSDK error code: The submitted handle is not a capability handle or was not initialized with GetDeviceCapabilitiesHandle (see page 162)().
	BS_SDK_ERROR_INVALID_INDEX (see page 463)	IsoSDK error code: The SubIndex of the Audiotrack is wrong.

	BS_SDK_ERROR_INVALID_ISRC (see page 464)	IsoSDK error code: The ISRC code inside the track information is invalid.
	BS_SDK_ERROR_INVALID_MCN (see page 464)	IsoSDK error code: The MCN code inside the track information is invalid.
	BS_SDK_ERROR_INVALID_PATH (see page 464)	IsoSDK error code: An invalid path has been passed in.
	BS_SDK_ERROR_INVALID_SESSION_NUMBER (see page 464)	IsoSDK error code: Invalid session number has been passed in.
	BS_SDK_ERROR_INVALID_SRC_PATH (see page 465)	IsoSDK error code: An invalid source path has been passed in.
	BS_SDK_ERROR_INVALID_UDF_VERSION (see page 465)	IsoSDK error code: Invalid UDF version number.
	BS_SDK_ERROR_INVALIDSRB (see page 465)	IsoSDK error code: Invalid SRB.
	BS_SDK_ERROR_ISOIMAGENOTFOUND (see page 465)	IsoSDK error code: Error open ISO image file. ISO image not found.
	BS_SDK_ERROR_MAXDIRS (see page 466)	IsoSDK error code: Not more than %d directories are allowed.
	BS_SDK_ERROR_MAXFILES (see page 466)	IsoSDK error code: Not more than %d files are allowed.
	BS_SDK_ERROR_MESSAGEJECT (see page 466)	IsoSDK error code: Message has been rejected.
	BS_SDK_ERROR_MISCOMPARE (see page 466)	IsoSDK error code: Miscompare. Compared names are different.
	BS_SDK_ERROR_MORE_SPACE_NEEDED (see page 467)	IsoSDK error code: The supplied buffer is too small.

	BS_SDK_ERROR_MP3LIB_NOT_FOUND (see page 467)	IsoSDK error code: mp3lib (FoxPlayer) not found.
	BS_SDK_ERROR_NETTAGS_CONNECT (see page 467)	IsoSDK error code: Error while connect to the CDDB server.
	BS_SDK_ERROR_NETTAGS_DISK (see page 467)	IsoSDK error code: Compact disc error during forming request to internet CDDB tags.
	BS_SDK_ERROR_NETTAGS_INTERNAL (see page 468)	IsoSDK error code: A internal error occurred. Please contact our support.
	BS_SDK_ERROR_NETTAGS_NOMATCH (see page 468)	IsoSDK error code: No match while searching CDDB database.
	BS_SDK_ERROR_NETTAGS_SERVER (see page 468)	IsoSDK error code: A server error occur. With FreeDB try another mirror.
	BS_SDK_ERROR_NEXTADDRESS (see page 468)	IsoSDK error code: Error getting next writable address.
	BS_SDK_ERROR_NO (see page 469)	IsoSDK error code: No error occurred.
	BS_SDK_ERROR_NOT_ALLOWED (see page 469)	IsoSDK error code: Operation / command not allowed.
	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER (see page 469)	IsoSDK error code: Not allowed for the current burner.
<span style="color: green;">new</span>	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE (see page 469)	IsoSDK error code: If you set a forbidden option or function according to the used project.
	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION (see page 470)	IsoSDK error code: Invalid option for this UDF version.
	BS_SDK_ERROR_NOT_IMPLEMENTED (see page 470)	IsoSDK error code: The feature ( command is currently not implemented.

	BS_SDK_ERROR_NOTREADY (see page 470)	IsoSDK error code: Device not ready.
	BS_SDK_ERROR_NOTSUPPORTED (see page 471)	IsoSDK error code: Operation / command not supported.
	BS_SDK_ERROR_PARITYERR (see page 471)	IsoSDK error code: Partition version error.
	BS_SDK_ERROR_PATH_EXISTS (see page 471)	IsoSDK error code: The path already exists.
	BS_SDK_ERROR_PLUGIN (see page 471)	IsoSDK error code: Error while loading external plugin.
	BS_SDK_ERROR_RESERVED (see page 472)	IsoSDK error code: Reserved.
	BS_SDK_ERROR_SELECTIONTIMEOUT (see page 472)	IsoSDK error code: Selection timed out.
	BS_SDK_ERROR_SRBTIMEOUT (see page 472)	IsoSDK error code: A SRB timeout occurred.
	BS_SDK_ERROR_TIMEOUT (see page 472)	IsoSDK error code: A timeout occurred.
	BS_SDK_ERROR_TOO MUCH DATA (see page 473)	IsoSDK error code: Too much data for this mode. The size of the collection will exceed the disc size.
	BS_SDK_ERROR_TOO MUCH INDEXES (see page 473)	IsoSDK error code: To much SubIndex values found. Max is 99.
	BS_SDK_ERROR_UNEXPECTEDBUSFREE (see page 473)	IsoSDK error code: Unexpected bus free.
	BS_SDK_ERROR_UNITATTENTION (see page 473)	IsoSDK error code: Unit attention.
	BS_SDK_ERROR_UNKNOWN (see page 474)	IsoSDK error code: An unknown error occurred.
	BS_SDK_ERROR_UNKNOWN_TEXTID (see page 474)	IsoSDK error code: The text ID that has been passed in is unknown.
	BS_SDK_ERROR_UNSUPPORTED_MEDIUM (see page 474)	IsoSDK error code: A medium was inserted that is not supported by IsoSDK.

	BS_SDK_ERROR_VOLUMEOVERFLOW (see page 474)	IsoSDK error code: Volume overflow.
	BS_SDK_INT_ERROR_1 (see page 475)	IsoSDK error code: Internal error 1.
	BS_SDK_INT_ERROR_2 (see page 475)	IsoSDK error code: Internal error 2.
	BS_SDK_INT_ERROR_3 (see page 475)	IsoSDK error code: Internal error 3.
	BS_SDK_INT_ERROR_4 (see page 475)	IsoSDK error code: Internal error 4.
	BS_SDK_INT_ERROR_5 (see page 476)	IsoSDK error code: Internal error 5.
	BS_SDK_INT_ERROR_FORMAT (see page 476)	IsoSDK error code: Error formatting re-writeable medium.
	BS_SDK_KEY_INVALID (see page 476)	IsoSDK message: An invalid license key was submitted.
	BS_SDK_KEY_VALID (see page 476)	IsoSDK message: The license key is valid.
	BS_SDK_MESSAGE_01 (see page 477)	IsoSDK Message: Text unknown
	BS_SDK_MESSAGE_02 (see page 477)	IsoSDK message: Waiting for user interaction
	BS_SDK_MESSAGE_03 (see page 477)	IsoSDK message: Cannot close dialog
	BS_SDK_MESSAGE_04 (see page 478)	IsoSDK message: Please stop the process first.
	BS_SDK_MESSAGE_05 (see page 478)	IsoSDK message: Erasing CD/DVD ...
	BS_SDK_MESSAGE_06 (see page 478)	IsoSDK message: Preparing Data ...
	BS_SDK_MESSAGE_07 (see page 479)	IsoSDK message: Finalizing CD/DVD ...

	BS_SDK_MESSAGE_08 (see page 479)	IsoSDK message: Burning CD/DVD ...
	BS_SDK_MESSAGE_10 (see page 479)	IsoSDK message: Aborting Process ...
	BS_SDK_MESSAGE_11 (see page 480)	IsoSDK message: Process successfully completed.
	BS_SDK_MESSAGE_12 (see page 480)	IsoSDK message: max.
	BS_SDK_MESSAGE_13 (see page 480)	IsoSDK message: Please insert the next CD/DVD.
	BS_SDK_MESSAGE_14 (see page 481)	IsoSDK message: Starting the verify process ...
	BS_SDK_MESSAGE_15 (see page 481)	IsoSDK message: Process completed with %d error(s).
	BS_SDK_MESSAGE_16 (see page 481)	IsoSDK message: Verifying file.
<b>new</b>	BS_SDK_MESSAGE_COMPENC_BOTH (see page 482)	The IsoSDK will throw this message if the signature for compression and encryption was found.
<b>new</b>	BS_SDK_MESSAGE_COMPENC_COMPRESSED (see page 482)	The IsoSDK will throw this message if the signature for compression was found.
<b>new</b>	BS_SDK_MESSAGE_COMPENC_ENCRYPTED (see page 482)	The IsoSDK will throw this message if the signature for encryption was found.
	BS_SDK_MESSAGE_ERASESTART (see page 482)	IsoSDK message: Starting the erase process.
	BS_SDK_MESSAGE_EXTR_FILE (see page 483)	IsoSDK message: Extracting file.
	BS_SDK_MESSAGE_FORMAT (see page 483)	IsoSDK message: Formatting.

	BS_SDK_MESSAGE_FORMAT_DONE ( <a href="#">see page 483</a> )	IsoSDK message: Formatting is done. Thrown after BS_SDK_MESSAGE_FORMAT ( <a href="#">see page 483</a> ).
	BS_SDK_MESSAGE_IMAGECREATESTART ( <a href="#">see page 484</a> )	IsoSDK message: Image creation started.
	BS_SDK_MESSAGE_IMPORT ( <a href="#">see page 484</a> )	IsoSDK message: Importing session.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY ( <a href="#">see page 484</a> )	IsoSDK message: Database entry can't be selected.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_MATCHING_PARAM ( <a href="#">see page 485</a> )	IsoSDK message:
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_PROVIDER ( <a href="#">see page 485</a> )	IsoSDK message: Error creating tag provider: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES ( <a href="#">see page 486</a> )	IsoSDK message: Error getting matching entries: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY ( <a href="#">see page 486</a> )	IsoSDK message: Error getting selected entry: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY ( <a href="#">see page 487</a> )	IsoSDK message: Fetching selected entry, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES ( <a href="#">see page 487</a> )	IsoSDK message: Getting matching entries, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS ( <a href="#">see page 488</a> )	IsoSDK message: Getting matching parameters, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE ( <a href="#">see page 488</a> )	IsoSDK message: Initialization complete. Processing...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES ( <a href="#">see page 489</a> )	IsoSDK message: Multiple matches.

	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY (see page 489)	IsoSDK message: Please, select an entry from the list below.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH (see page 490)	IsoSDK message: Single match.
	BS_SDK_MESSAGE_SIMULATE (see page 490)	IsoSDK message: Test write.
	BS_SDK_MESSAGE_WAIT (see page 490)	IsoSDK message: Writing lead-out (may take 15-20 minutes).
	BS_SDK_MESSAGE_WRITESTART (see page 491)	IsoSDK message: Starting write process.
	BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE (see page 491)	IsoSDK error code: The burned file is not readable.
	BS_SDK_VERIFY_ERROR_FILESDIFFERENT (see page 491)	IsoSDK error code: The burned file differs from the source file.
	BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE (see page 491)	IsoSDK error code: The source file is not readable.
	BS_SEPARATE_CHANNEL_MUTE (see page 492)	Drive feature: The Logical Unit shall support independently muting each audio channel via the CD Audio Control Page.
	BS_SEPARATE_VOLUME_LEVELS (see page 492)	Drive feature: The Logical Unit shall support separately controllable audio levels for each supported channel via the CD Audio Control Page.
	BS_SMART (see page 492)	Drive feature: Device is able to perform Self-Monitoring Analysis and Reporting Technology.

	BS_STREAMING (see page 492)	Drive feature: Device is able to perform reading and writing within specified performance ranges.
<span style="background-color: #e0f2e0; border: 1px solid black; padding: 2px;">new</span>	BS_STRING_MANIPULATION_DETECTED (see page 493)	The IsoSDK will check the string length inside the function body of dedicated classes. If a wrong string length was found the IsoSDK will assume a manipulation and will stop to avoid attacks.
	BS_TCH_CDTEXT (see page 493)	Audio grabber tag information: Receive tags only from CD-Text.
	BS_TCH_CDTEXT_INTERNETDB (see page 493)	Audio grabber tag information: Receive tags first from CD-Text then from FreeDB / CDDB.
	BS_TCH_INTERNETDB (see page 493)	Audio grabber tag information: Receive tags only from FreeDB / CDDB
	BS_TCH_INTERNETDB_CDTEXT (see page 494)	Audio grabber tag information: Receive tags First from FreeDB/CDDB then from CD-Text.
	BS_TCH_NONE (see page 494)	Audio grabber tag information: Receive no tags.
	BS_TCI_ARTIST (see page 494)	Audio disc CDDB tag information: Artist information.
	BS_TCI_CATEGORY (see page 494)	Audio disc CDDB tag information: Category information.
	BS_TCI_DISK_LENGTH (see page 495)	Audio disc CDDB tag information: disc length information

	BS_TCI_DISKID (see page 495)	Audio disc CDDB tag information: Get the Disc id.
	BS_TCI_EXTENDED_INFO (see page 495)	Audio disc CDDB tag information: Extended information.
	BS_TCI_FRAME_OFFSET (see page 495)	Audio disc CDDB tag information: Frame offset information.
	BS_TCI_GENRE (see page 496)	Audio disc CDDB tag information: Genre information.
	BS_TCI_REVISION (see page 496)	Audio disc CDDB tag information: Revision information.
	BS_TCI_SUBMITTED_VIA (see page 496)	Audio disc CDDB tag information: Receive information about the identifier that was used to submit data.
	BS_TCI_TITLE (see page 496)	Audio disc CDDB tag information: Receive track or disc title.
	BS_TCI_YEAR (see page 497)	Audio disc CDDB tag information: Receive production year.
	BS_TF_AUDIO (see page 497)	Track type: Audio track.
	BS_TF_DATA_MODE1 (see page 497)	Track type: Mode 1 track (always for DVD/BD).
	BS_TF_DATA_MODE2 (see page 497)	Track type: Mode 2 track.
	BS_TM_DMA (see page 498)	IsoSDK transfer mode: Direct Memory Access
	BS_TM_NOT_APPLICABLE (see page 498)	IsoSDK transfer mode: Not possible to evaluate or set.

	BS_TM_PIO (see page 498)	IsoSDK transfer mode: Programmed Input/Output
	BS_TM_UNKNOWN (see page 498)	IsoSDK transfer mode: Unknown
	BS_TRUE (see page 499)	IsoSDK BOOL value: TRUE
	BS_UDF_PARTITION_PHYSICAL (see page 499)	UDF partition type: Physical or Type 1 This is the simplest partition. A type 1 partition has a start address S and size N. A logical block number A in the partition can be converted to the media physical address (in UDF's term, the logical sector address) S+A. In certain optical media, the start and size of the partition must be aligned to the packet size (such as 32KB). These special requirements are defined in the appendixes of the UDF standard. Free space of the partition is managed by the Unallocated Space Bitmap Descriptor. It contains one bit for each... more (see page 499)

	BS_UDF_PARTITION_SPARABLE (see page 499)	UDF partition type: Sparable. Sparable partitions are used on overwrite media that will fail after a certain number of overwrites (several thousands), such as CD-RW. In a file system, the places that are overwritten frequently are often important metadata area, e.g., bitmaps. Sparable partition allows the failed area to be remapped to other good part on the media so the failed area appears good to the upper level. A sparable partition is similar to a type 1 partition in the sense that it has a start address and size. Moreover, it defines 2 to 4 sparing tables which points to... more (see page 499)
--	--	---

	BS_UDF_PARTITION_VIRTUAL (see page 500)	UDF partition type: Virtual. Virtual partition is used on write-once media. Only three types of metadata are stored in the virtual partition: File Set Descriptor, File Entry (including Extended File Entry), and Allocation Extent Descriptor. If the file data is embedded in the file entry, these file data are also stored in the virtual partition. Virtual partition makes the write-once media appear as an overwrite media. Virtual partition layers on top of the type 1 partition. A Virtual Allocation Table (VAT) is used to map logical addresses of the virtual partition to logical addresses in the underlying type 1 partition.... more (see page 500)
	BS_UDF_VERSION_102 (see page 500)	UDF version: 1.02 - VideoDVD
	BS_UDF_VERSION_150 (see page 500)	UDF version: 1.50
	BS_UDF_VERSION_200 (see page 501)	UDF version: 2.00
	BS_UDF_VERSION_201 (see page 501)	UDF version: 2.01
	BS_UDF_VERSION_250 (see page 501)	UDF version: 2.50
	BS_UDF_VERSION_260 (see page 501)	UDF version: 2.60
	BS_UDFOPT_ALL (see page 502)	UDF options: Set all options.
	BS_UDFOPT_AVCHD_DISC (see page 502)	UDF options: Set AVCHD flag.
	BS_UDFOPT_FILE_STREAMS (see page 502)	UDF options: Set write file streams.

	BS_UDFOPT_IMPLEMENTATION_ID (see page 502)	UDF options: Write implementation ID.
	BS_UDFOPT_PARTITION_TYPE (see page 503)	UDF options: Write partition type.
	BS_UDFOPT_VERSION (see page 503)	UDF options: Write UDF version.
	BS_UNDERRUN_PROTECTION (see page 503)	Drive feature: Device is able to perform buffer protection functions like <b>Burn-Proof</b> .
	BS_UNKNOWN (see page 503)	A disc identifier for disc type: Unknown
	BS_UPC_READ (see page 504)	Drive feature: Device is able to read the UPC info of a disc.
	BS_VCD_INFINITE (see page 504)	This is macro BS_VCD_INFINITE.
	BS_VCD_KEY_0 (see page 504)	This is macro BS_VCD_KEY_0 .
	BS_VCD_KEY_DEFAULT (see page 504)	This is macro BS_VCD_KEY_DEFAULT.
	BS_VCD_KEY_NEXT (see page 505)	This is macro BS_VCD_KEY_NEXT.
	BS_VCD_KEY_PREVIOUS (see page 505)	This is macro BS_VCD_KEY_PREVIOUS.
	BS_VCD_KEY_RETURN (see page 505)	This is macro BS_VCD_KEY_RETURN.
	BS_VCD_SEGMENT_ITEM_0 (see page 505)	This is macro BS_VCD_SEGMENT_ITEM_0.
	BS_VCD_TRACK_ITEM_0 (see page 506)	This is macro BS_VCD_TRACK_ITEM_0.
	BS_WHOLE_PATH (see page 506)	The complete path of the file is created as a subfolder in the chDestinationPath of the project.
	BS_WM_DAO (see page 506)	Write method: DAO (Disc-At-Once)

	BS_WM.DAO96 (see page 506)	Write method: DAO96 (Disc-At-Once + 96)
	BS_WM.TAO (see page 507)	Write method: TAO (Track-At-Once)
	BS_WRITE_BLURAY_R (see page 507)	Drive feature: Device is able to write Blu-Ray Recordable.
	BS_WRITE_BLURAY_R_XL (see page 507)	Drive feature: Device is able to write BD-R XL.
	BS_WRITE_BLURAY_RE (see page 507)	Drive feature: Device is able to write Blu-Ray ReWritable.
	BS_WRITE_BLURAY_RE_XL (see page 508)	Drive feature: Device is able to write BD-RE XL (Re-Writable).
	BS_WRITE_CDR (see page 508)	Drive feature: Device is able to write CD-Recordable.
	BS_WRITE_CDRW (see page 508)	Drive feature: Device is able to write CD-ReWritable.
	BS_WRITE_CDRW_CAV (see page 508)	Drive feature: Device is able to write CD-RW media that is designed for CAV recording.
	BS_WRITE_DVD_DL (see page 509)	Drive feature: Device is able to write DVD-R Double Layer.
	BS_WRITE_DVD_MRDL (see page 509)	Drive feature: Device is able to write DVD-R Double Layer.
	BS_WRITE_DVD_RDL_PLUS (see page 509)	Drive feature: Device is able to write DVD+R Double Layer.
	BS_WRITE_DVD_RWDL_PLUS (see page 509)	Drive feature: Device is able to write DVD+ReWritable Double Layer.

	BS_WRITE_DVDR (see page 510)	Drive feature: Device is able to write DVD-R.
	BS_WRITE_DVDR_PLUS (see page 510)	Drive feature: Device is able to write DVD+R.
	BS_WRITE_DVDRAM (see page 510)	Drive feature: Device is able to write DVD-RAM.
	BS_WRITE_DVDRW (see page 510)	Drive feature: Device is able to write DVD-ReWritable.
	BS_WRITE_DVDRW_PLUS (see page 511)	Drive feature: Device is able to write DVD+ReWritable.
	BS_WRITE_HDDVD_R (see page 511)	Drive feature: Device is able to write HDDVD Reordable.
	BS_WRITE_HDDVD_RW (see page 511)	Drive feature: Device is able to write HDDVD ReWritable.
	BS_WRITE_MOUNT_RAINER (see page 511)	Drive feature: Device is able to write Mount Rainer.
	BS_WRITE_TEST (see page 512)	Drive feature: Device is able to simulate burning.
	COMPRESS_BLOCKSIZE (see page 512)	This is macro COMPRESS_BLOCKSIZE.
	ENCRYPT_BLOCKSIZE (see page 512)	This is macro ENCRYPT_BLOCKSIZE.
	FOpen (see page 512)	This is macro FOpen.
	INSERT_STRUCTURE_PADDING (see page 513)	This is macro INSERT_STRUCTURE_PADDING.
	int2bool (see page 513)	This is macro int2bool.
	int2BS_BOOL (see page 513)	This is macro int2BS_BOOL.
	PATHSEPSTRING (see page 513)	This is macro PATHSEPSTRING.
	SPrintf (see page 514)	This is macro SPrintf.
	StrCmp (see page 514)	This is macro StrCmp.
	StrCpy (see page 514)	This is macro StrCpy.

	StrLen ( <a href="#">see page 514</a> )	This is macro StrLen.
	StrnSet ( <a href="#">see page 515</a> )	This is macro StrnSet.
	STRUCTURE_PADDING_BIG ( <a href="#">see page 515</a> )	This is macro STRUCTURE_P ADDING_BIG.
	STRUCTURE_PADDING_HUGE ( <a href="#">see page 515</a> )	This is macro STRUCTURE_P ADDING_HUGE.
	STRUCTURE_PADDING_NORMAL ( <a href="#">see page 515</a> )	This is macro STRUCTURE_P ADDING_NORMAL.
	STRUCTURE_PADDING_SMALL ( <a href="#">see page 516</a> )	This is macro STRUCTURE_P ADDING_SMALL.
	STRUCTURE_PADDING_STRING_DATA ( <a href="#">see page 516</a> )	This is macro STRUCTURE_P ADDING_STRING_DATA.
	VsPrintf ( <a href="#">see page 516</a> )	This is macro VsPrintf.

## 1.5.1 \_\_\_\_stat64 Macro

### C++

```
#define ____stat64 struct _stat64
```

### File

IsoSDKDefinitions.h ( [see page 517](#) )

### Description

This is macro \_\_\_\_stat64.

## 1.5.2 \_\_BS\_LIBRARY\_H\_\_ Macro

### C++

```
#define __BS_LIBRARY_H__
```

### File

IsoSDKBurningLib.h ( [see page 517](#) )

### Description

This is macro \_\_BS\_LIBRARY\_H\_\_.

## 1.5.3 \_\_ISOSDK\_DEFINITIONS\_H\_\_ Macro

### C++

```
#define __ISOSDK_DEFINITIONS_H__
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro \_\_ISOSDK\_DEFINITIONS\_H\_\_.

---

## 1.5.4 \_\_ISOSDK\_EXPORT\_H\_\_ Macro

### C++

```
#define __ISOSDK_EXPORT_H__
```

### File

IsoSDKExport.h ([see page 577](#))

### Description

This is macro \_\_ISOSDK\_EXPORT\_H\_\_.

---

## 1.5.5 \_\_max Macro

### C++

```
#define __max(a,b) (((a) > (b)) ? (a) : (b))
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro \_\_max.

---

## 1.5.6 \_\_min Macro

### C++

```
#define __min(a,b) (((a) < (b)) ? (a) : (b))
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro \_\_min.

## 1.5.7 \_FindData\_t Macro

**C++**

```
#define _FindData_t _finddata_t
```

**File**

IsoSDKUnicode.h ([see page 587](#))

**Description**

This is macro \_FindData\_t.

---

## 1.5.8 \_FindFirst Macro

**C++**

```
#define _FindFirst _findfirst
```

**File**

IsoSDKUnicode.h ([see page 587](#))

**Description**

This is macro \_FindFirst.

---

## 1.5.9 \_MAX\_PATH Macro

**C++**

```
#define _MAX_PATH 0x8000
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro \_MAX\_PATH.

---

## 1.5.10 BOOL2bool Macro

**C++**

```
#define BOOL2bool(bTmp) (FALSE != bTmp)
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BOOL2bool.

## 1.5.11 BS\_ANALOG\_AUDIO\_PLAYBACK Macro

### C++

```
#define BS_ANALOG_AUDIO_PLAYBACK 0xFFFF00000001LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device support playback analog audio.

---

## 1.5.12 BS\_API Macro

### C++

```
#define BS_API __declspec(dllimport)
```

### File

IsoSDKBurningLib.h (see page 517)

### Description

VideoCD / SuperVideoCD - MPEG file has a 16:9 display

---

## 1.5.13 BS\_AR\_16TO9\_DISPLAY Macro

### C++

```
#define BS_AR_16TO9_DISPLAY 2
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Video CD / Super Video CD - MPEG file has a 16:9 display.

---

## 1.5.14 BS\_AR\_221TO2\_DISPLAY Macro

### C++

```
#define BS_AR_221TO2_DISPLAY 3
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Video CD / Super Video CD - MPEG file has a 2.21:1 display.

## 1.5.15 BS\_AR\_4TO3\_DISPLAY Macro

**C++**

```
#define BS_AR_4TO3_DISPLAY 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Video CD / Super Video CD - The MPEG file has a 4:3 display.

---

## 1.5.16 BS\_AR\_SQUARE\_PIXELS Macro

**C++**

```
#define BS_AR_SQUARE_PIXELS 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Video CD / Super Video CD - MPEG file has square pixels aspect ratio.

---

## 1.5.17 BS\_AR\_UNKNOWN Macro

**C++**

```
#define BS_AR_UNKNOWN 4
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Video CD / Super Video CD - MPEG file has an unknown aspect ratio.

---

## 1.5.18 BS\_ASPI\_FROGASPI Macro

**C++**

```
#define BS_ASPI_FROGASPI 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

The FrogAspi DLL. This is a very good replacement of the Adaptec interface. It supports all new interfaces.

**Notes**

The FrogAspi is unfortunately no longer available. On modern operating systems like Windows XP and higher FrogAspi

should no longer be used.

---

## 1.5.19 BS\_ASPI\_INTERNAL Macro

**C++**

```
#define BS_ASPI_INTERNAL 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Use the IsoSDK internal ASPI. This is recommended for all Linux and macOS versions and Windows from Windows 2000 and above.

---

## 1.5.20 BS\_ASPI\_WNASPI Macro

**C++**

```
#define BS_ASPI_WNASPI 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

The default external ASPI layer like the one from Adaptec. Please note that this ASPI does neither support USB nor FireWire.

---

## 1.5.21 BS\_AUDIO\_MP3 Macro

**C++**

```
#define BS_AUDIO_MP3 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio format is MP3 (MPEG Audio Layer 3).

---

## 1.5.22 BS\_AUDIO\_NO Macro

**C++**

```
#define BS_AUDIO_NO 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio format is unkown.

## 1.5.23 BS\_AUDIO\_OGG Macro

**C++**

```
#define BS_AUDIO_OGG 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio format is [Ogg Vorbis](#).

---

## 1.5.24 BS\_AUDIO\_PCM Macro

**C++**

```
#define BS_AUDIO_PCM 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio format is PCM.

---

## 1.5.25 BS\_BARCODE\_READ Macro

**C++**

```
#define BS_BARCODE_READ 0xFFFF0000000ELL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read the disc barcode information.

---

## 1.5.26 BS\_BLURAY\_R Macro

**C++**

```
#define BS_BLURAY_R 18
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: Blu-ray BD-R

## 1.5.27 BS\_BLURAY\_R\_RRM Macro

**C++**

```
#define BS_BLURAY_R_RRM 21
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: Blu-ray BD-RRM

---

## 1.5.28 BS\_BLURAY\_RE Macro

**C++**

```
#define BS_BLURAY_RE 19
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: Blu-ray BD-RE

---

## 1.5.29 BS\_BLURAY\_ROM Macro

**C++**

```
#define BS_BLURAY_ROM 20
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: Blu-ray BD-ROM

---

## 1.5.30 BS\_BOOL Macro

**C++**

```
#define BS_BOOL int8
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_BOOL.

## 1.5.31 BS\_BOOL2bool Macro

**C++**

```
#define BS_BOOL2bool(bTmp) (BS_FALSE != bTmp)
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio format is MP3 (MPEG Audio Layer 3).

---

## 1.5.32 BS\_BT\_AVERAGE Macro

**C++**

```
#define BS_BT_AVERAGE 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio encoding: Average [Bitrate \(CBR\)](#)

---

## 1.5.33 BS\_BT\_CONSTANT Macro

**C++**

```
#define BS_BT_CONSTANT 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio encoding: Constant [Bitrate \(CBR\)](#)

---

## 1.5.34 BS\_BT\_VARIABLE Macro

**C++**

```
#define BS_BT_VARIABLE 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio encoding: Variable [Bitrate \(VBR\)](#)

## 1.5.35 BS\_BURN\_DEVICE Macro

### C++

```
#define BS_BURN_DEVICE BS_CURRENT_DEVICE
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK device type: Burn ([see page 121](#)) Device.

---

## 1.5.36 BS\_BURNER\_AUDIO Macro

### C++

```
#define BS_BURNER_AUDIO 0
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK project type AudioCD.

---

## 1.5.37 BS\_BURNER\_BLURAY Macro

### C++

```
#define BS_BURNER_BLURAY 8
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK project type Blu-Ray disc. This project type is for Blu-ray Video. It do not contain a ISO bridge. For ISO/UDF Blu-ray please use BS\_BURNER\_MIXEDMODE ([see page 315](#)) project type.

---

## 1.5.38 BS\_BURNER\_CUE Macro

### C++

```
#define BS_BURNER_CUE 1
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK project type Bin/Cue Image.

## 1.5.39 BS\_BURNER\_DATA Macro

**C++**

```
#define BS_BURNER_DATA 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK project type Data CD/DVD. ISO9660 file system + extension.

---

## 1.5.40 BS\_BURNER\_ISOUDF Macro

**C++**

```
#define BS_BURNER_ISOUDF 7
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK project type Data CD/DVD with ISO UDF Bridge.

---

## 1.5.41 BS\_BURNER\_MIXEDMODE Macro

**C++**

```
#define BS_BURNER_MIXEDMODE 9
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK project type Mixed Mode CD (Audio/Data).

---

## 1.5.42 BS\_BURNER\_RAW Macro

**C++**

```
#define BS_BURNER_RAW 10
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK project type RAW Image (Universal). This project type can burn any type of disc images until you know the details.

## 1.5.43 BS\_BURNER\_SVCD Macro

**C++**

```
#define BS_BURNER_SVCD 4
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK project type SuperVideoCD.

---

## 1.5.44 BS\_BURNER\_UDFDVD Macro

**C++**

```
#define BS_BURNER_UDFDVD 6
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK project type Data CD/DVD with UDF only.

---

## 1.5.45 BS\_BURNER\_VCD Macro

**C++**

```
#define BS_BURNER_VCD 3
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK project type VideoCD.

---

## 1.5.46 BS\_BURNER\_VIDEOODVD Macro

**C++**

```
#define BS_BURNER_VIDEOODVD 5
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK type VideoDVD (ISO/UDF 1.02). This project type contain folders for VideoDVD and special sorting features for VideoDVD.

## 1.5.47 BS\_C2\_POINTERS Macro

**C++**

```
#define BS_C2_POINTERS 0xFFFFF0000000BLL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device support reading C2 error pointers and C2 block error flags.

---

## 1.5.48 BS\_CALL Macro

**C++**

```
#define BS_CALL
```

**File**

IsoSDKExport.h (see page 577)

**Description**

This is macro BS\_CALL.

---

## 1.5.49 BS\_CD\_R Macro

**C++**

```
#define BS_CD_R 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: CD-R

---

## 1.5.50 BS\_CD\_ROM Macro

**C++**

```
#define BS_CD_ROM 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: CD-ROM

## 1.5.51 BS\_CD\_RW Macro

**C++**

```
#define BS_CD_RW 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: CD-ReWritable

---

## 1.5.52 BS\_CD\_TEXT\_READ Macro

**C++**

```
#define BS_CD_TEXT_READ 0xFFFF0000001ALL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read the CD-TEXT information of a disc (subchannel).

---

## 1.5.53 BS\_CD\_TEXT\_WRITE Macro

**C++**

```
#define BS_CD_TEXT_WRITE 0xFFFF0000001BLL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write the CD-TEXT information of a disc (subchannel).

---

## 1.5.54 BS\_CDDA\_COMMANDS Macro

**C++**

```
#define BS_CDDA_COMMANDS 0xFFFF00000007LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to work with CDDA command set.

---

## 1.5.55 BS\_CDDA\_STREAM\_IS\_ACCURATE Macro

### C++

```
#define BS_CDDA_STREAM_IS_ACCURATE 0xFFFF00000008LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit shall support an audio location without losing place to continue the READ CD command.

---

## 1.5.56 BS\_CDTCI\_ARRANGER Macro

### C++

```
#define BS_CDTCI_ARRANGER 0
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

CD-Text content items: Arranger field.

---

## 1.5.57 BS\_CDTCI\_COMPOSER Macro

### C++

```
#define BS_CDTCI_COMPOSER 1
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

CD-Text content items: Composer field.

---

## 1.5.58 BS\_CDTCI\_MESSAGE Macro

### C++

```
#define BS_CDTCI_MESSAGE 3
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

CD-Text content items: Message field.

---

## 1.5.59 BS\_CDTCI\_PERFORMER Macro

**C++**

```
#define BS_CDTCI_PERFORMER 4
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

CD-Text content items: Performer field.

---

## 1.5.60 BS\_CDTCI\_SONG\_WRITER Macro

**C++**

```
#define BS_CDTCI_SONG_WRITER 5
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

CD-Text content items: Songwriter field.

---

## 1.5.61 BS\_CDTCI\_TITLE Macro

**C++**

```
#define BS_CDTCI_TITLE 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

CD-Text content items: Title field.

---

## 1.5.62 BS\_CHANGER\_SIDE\_CHANGE\_CAPABLE Macro

**C++**

```
#define BS_CHANGER_SIDE_CHANGE_CAPABLE 0xFFFF00000017LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: The Logical Unit is capable of selecting both sides of discs.

## 1.5.63 BS\_CHANGER\_SOFTWARE\_SLOT\_SELECTION Macro

### C++

```
#define BS_CHANGER_SOFTWARE_SLOT_SELECTION 0xFFFF00000016LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: This bit controls the behavior of the LOAD/UNLOAD MEDIUM command when trying to load a Slot with no disc present.

## 1.5.64 BS\_CHANGER\_SUPPORTS\_DISC\_PRESENT Macro

### C++

```
#define BS_CHANGER_SUPPORTS_DISC_PRESENT 0xFFFF00000015LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit contains an embedded changer, and after a reset condition or a magazine change, the Logical Unit is capable of reporting the exact contents of the slots.

## 1.5.65 BS\_COMPOSITE\_AUDIO\_AND\_VIDEO Macro

### C++

```
#define BS_COMPOSITE_AUDIO_AND_VIDEO 0xFFFF00000002LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit shall be capable of delivering a composite Audio and Video data stream from an independent digital port.

## 1.5.66 BS\_COMPRESSED\_SIGNATURE Macro

### C++

```
#define BS_COMPRESSED_SIGNATURE "FoxC"
```

### File

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_COMPRESSED\_SIGNATURE.

---

## 1.5.67 BS\_COMPRESSED\_SIGNATURE\_FOR\_DRIVER Macro

**C++**

```
#define BS_COMPRESSED_SIGNATURE_FOR_DRIVER 'CxoF'
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_COMPRESSED\_SIGNATURE\_FOR\_DRIVER.

---

## 1.5.68 BS\_CONTINUE\_LAST\_SESSION Macro

**C++**

```
#define BS_CONTINUE_LAST_SESSION 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Default value to import last session of the inserted disc.

---

## 1.5.69 BS\_CONTINUE\_NO\_SESSION Macro

**C++**

```
#define BS_CONTINUE_NO_SESSION -1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Default value to do not import any available session.

---

## 1.5.70 BS\_CPRMAUTH Macro

**C++**

```
#define BS_CPRMAUTH 0x010000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Device is able to CPRM authentication and key management.

---

## 1.5.71 BS\_CURRENT\_DEVICE Macro

**C++**

```
#define BS_CURRENT_DEVICE (-1)
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Default device index of the current selected drive.

---

## 1.5.72 BS.DAO\_16 Macro

**C++**

```
#define BS.DAO_16 0xFFFF0000001DLL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device can write DAO16 method.

---

## 1.5.73 BS.DAO\_96\_PACK Macro

**C++**

```
#define BS.DAO_96_PACK 0xFFFF0000001ELL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device can write DAO96 packet write method.

---

## 1.5.74 BS.DAO\_96\_RAW Macro

**C++**

```
#define BS.DAO_96_RAW 0xFFFF0000001FLL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device can write DAO96 raw write method.

---

## 1.5.75 BS.DAO\_RAW Macro

**C++**

```
#define BS.DAO_RAW 0xFFFF0000001CLL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device can write DAO raw write method.

---

## 1.5.76 BS.DDCD\_R Macro

**C++**

```
#define BS.DDCD_R 13
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

A disc identifier for disc type: Double Density Compact Disc - Recording

---

## 1.5.77 BS.DDCD\_ROM Macro

**C++**

```
#define BS.DDCD_ROM 12
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

A disc identifier for disc type: Double Density Compact Disc - ROM

---

## 1.5.78 BS.DDCD\_RW Macro

**C++**

```
#define BS.DDCD_RW 14
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

A disc identifier for disc type: Double Density Compact Disc - ReWritable

---

## 1.5.79 BS\_DEFAULT\_LANGUAGE\_FILE Macro

### C++

```
#define BS_DEFAULT_LANGUAGE_FILE _T( "Burning.lng" )
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The defined default language file name: "language.ini".

---

## 1.5.80 BS\_DEFAULT\_REGISTRY\_KEY Macro

### C++

```
#define BS_DEFAULT_REGISTRY_KEY _T( "HKEY_CURRENT_USER" )
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The defined default registry key: "HKEY\_CURRENT\_USER".

---

## 1.5.81 BS\_DEFAULT\_REGISTRY\_PATH Macro

### C++

```
#define BS_DEFAULT_REGISTRY_PATH _T( "\\Software\\IsoSDK\\" )
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The default defined registry path: "\\Software\\IsoSDK\\\".

---

## 1.5.82 BS\_DEFECTMANAGEMENT Macro

### C++

```
#define BS_DEFECTMANAGEMENT 0x020000000000LL
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

Drive feature: Device has defect management feature. Available to provide defect-free address space. For example BD recorders provide this feature.

## 1.5.83 BS\_DIGITAL\_PORT\_1 Macro

**C++**

```
#define BS_DIGITAL_PORT_1 0xFFFF00000003LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support digital output (IEC958) on port 1.

---

## 1.5.84 BS\_DIGITAL\_PORT\_2 Macro

**C++**

```
#define BS_DIGITAL_PORT_2 0xFFFF00000004LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support digital output (IEC958) on port 2.

---

## 1.5.85 BS\_DONT\_SAVE\_PATH Macro

**C++**

```
#define BS_DONT_SAVE_PATH 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

The file is created in the chDestinationPath of the project.

---

## 1.5.86 BS\_DVD\_MRDL Macro

**C++**

```
#define BS_DVD_MRDL 17
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-R Double Layer

## 1.5.87 BS\_DVD\_PLUSR Macro

**C++**

```
#define BS_DVD_PLUSR 11
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD+R

---

## 1.5.88 BS\_DVD\_PLUSRW Macro

**C++**

```
#define BS_DVD_PLUSRW 10
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD+R ReWritable

---

## 1.5.89 BS\_DVD\_R Macro

**C++**

```
#define BS_DVD_R 5
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-R

---

## 1.5.90 BS\_DVD\_RAM Macro

**C++**

```
#define BS_DVD_RAM 6
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-RAM

## 1.5.91 BS\_DVD\_RDL\_PLUS Macro

**C++**

```
#define BS_DVD_RDL_PLUS 15
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD+R Double Layer

---

## 1.5.92 BS\_DVD\_ROM Macro

**C++**

```
#define BS_DVD_ROM 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-ROM

---

## 1.5.93 BS\_DVD\_RW Macro

**C++**

```
#define BS_DVD_RW 8
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-R ReWritable

---

## 1.5.94 BS\_DVD\_RW\_RO Macro

**C++**

```
#define BS_DVD_RW_RO 7
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-R ReWritable - Restricted Overwrite

## 1.5.95 BS\_DVD\_RW\_SR Macro

**C++**

```
#define BS_DVD_RW_SR 9
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD-R ReWritable - Sequential Recording

---

## 1.5.96 BS\_DVD\_RWDL\_PLUS Macro

**C++**

```
#define BS_DVD_RWDL_PLUS 16
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: DVD+R ReWritable Double Layer

---

## 1.5.97 BS\_EJECT\_INDIVIDUAL\_OR\_MAGAZINE Macro

**C++**

```
#define BS_EJECT_INDIVIDUAL_OR_MAGAZINE 0xFFFF00000012LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support media eject via the START STOP UNIT command with the LoEj bit set.

---

## 1.5.98 BS\_EMT\_BD Macro

**C++**

```
#define BS_EMT_BD 7
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: Blu-Ray medium

## 1.5.99 BS\_EMT\_CD\_AUDIO Macro

**C++**

```
#define BS_EMT_CD_AUDIO 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: [CD-DA](#) format

---

## 1.5.100 BS\_EMT\_CD\_ENHANCED Macro

**C++**

```
#define BS_EMT_CD_ENHANCED 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: [CD-Enhanced](#) format

---

## 1.5.101 BS\_EMT\_CD\_MIXED\_MODE Macro

**C++**

```
#define BS_EMT_CD_MIXED_MODE 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: [Mixed Mode](#) CD format

---

## 1.5.102 BS\_EMT\_CD\_MULTISESSION Macro

**C++**

```
#define BS_EMT_CD_MULTISESSION 5
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: Multi-session data CD

## 1.5.103 BS\_EMT\_CD\_ROM Macro

**C++**

```
#define BS_EMT_CD_ROM 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: Data CD or CD of unknown format

---

## 1.5.104 BS\_EMT\_CD\_ROM\_XA Macro

**C++**

```
#define BS_EMT_CD_ROM_XA 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: CD-eXtended Architecture format

---

## 1.5.105 BS\_EMT\_DVD Macro

**C++**

```
#define BS_EMT_DVD 6
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: DVD medium

---

## 1.5.106 BS\_EMT\_HDDVD Macro

**C++**

```
#define BS_EMT_HDDVD 8
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium type: HDDVD medium

## 1.5.107 BS\_ET\_AAC Macro

**C++**

```
#define BS_ET_AAC 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: AAC ([Advanced Audio Codec](#))

---

## 1.5.108 BS\_ET\_FLAC Macro

**C++**

```
#define BS_ET_FLAC 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: FLAC ([Free Lossless Audio Codec](#)).

---

## 1.5.109 BS\_ET\_MP3 Macro

**C++**

```
#define BS_ET_MP3 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: MP3 ([MPEG Audio Layer 3](#)).

---

## 1.5.110 BS\_ET\_MP4 Macro

**C++**

```
#define BS_ET_MP4 6
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: MEPEG-4 ([Mp4](#))

## 1.5.111 BS\_ET\_OGG Macro

**C++**

```
#define BS_ET_OGG 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: [Ogg Vorbis](#)

---

## 1.5.112 BS\_ET\_OPUS Macro

**C++**

```
#define BS_ET_OPUS 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: [Opus](#)

---

## 1.5.113 BS\_ET\_WMA Macro

**C++**

```
#define BS_ET_WMA 5
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Encoding type: WMA (Windows Media Audio)

---

## 1.5.114 BS\_FA\_ADVANCED\_HIDDEN Macro

**C++**

```
#define BS_FA_ADVANCED_HIDDEN 0x00000040
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File attribute: Advanced hidden files. Not valid in all OS. Use in combination with ISO files.

## 1.5.115 BS\_FA\_ALL Macro

**C++**

```
#define BS_FA_ALL 0x0000007F
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File attribute: All attributes set.

---

## 1.5.116 BS\_FA\_ARCHIVE Macro

**C++**

```
#define BS_FA_ARCHIVE 0x00000020
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File attribute: Archive file.

---

## 1.5.117 BS\_FA\_DIRECTORY Macro

**C++**

```
#define BS_FA_DIRECTORY 0x00000010
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File attribute: Folder / Directory.

---

## 1.5.118 BS\_FA\_HIDDEN Macro

**C++**

```
#define BS_FA_HIDDEN 0x00000002
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File attribute: Hidden file.

## 1.5.119 BS\_FA\_READONLY Macro

**C++**

```
#define BS_FA_READONLY 0x00000001
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

File attribute: Write protected files.

---

## 1.5.120 BS\_FA\_SYSTEM Macro

**C++**

```
#define BS_FA_SYSTEM 0x00000004
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

File attribute: System files.

---

## 1.5.121 BS\_FALSE Macro

**C++**

```
#define BS_FALSE 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK BOOL value: FALSE

---

## 1.5.122 BS\_FF\_BIN Macro

**C++**

```
#define BS_FF_BIN 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

File format is BIN (2352).

## 1.5.123 BS\_FF\_ISO Macro

**C++**

```
#define BS_FF_ISO 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File format is ISO (2048)

---

## 1.5.124 BS\_FF\_MPEG Macro

**C++**

```
#define BS_FF_MPEG 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File format is MPEG.

---

## 1.5.125 BS\_FF\_WAVE Macro

**C++**

```
#define BS_FF_WAVE 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

File format is WAVE (PCM).

---

## 1.5.126 BS\_FS\_BOOTABLE Macro

**C++**

```
#define BS_FS_BOOTABLE 8
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc file system: El Torito bootable extension to ISO 9660.

## 1.5.127 BS\_FS\_ISO9660 Macro

**C++**

```
#define BS_FS_ISO9660 1
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

Disc file system: ISO 9660.

---

## 1.5.128 BS\_FS\_JOLIET Macro

**C++**

```
#define BS_FS_JOLIET 2
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

Disc file system: [Joliet](#) extension to ISO 9660.

---

## 1.5.129 BS\_FS\_ROCKRIDGE Macro

**C++**

```
#define BS_FS_ROCKRIDGE 16
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

Disc file system: [Rock Ridge](#).

---

## 1.5.130 BS\_FS\_UDF Macro

**C++**

```
#define BS_FS_UDF 4
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

Disc file system: [Universal Disk Format](#).

## 1.5.131 BS\_FS\_UNKNOWN Macro

**C++**

```
#define BS_FS_UNKNOWN 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc file system: No file system or unknown file system.

---

## 1.5.132 BS\_GUI\_ERROR\_01 Macro

**C++**

```
#define BS_GUI_ERROR_01 776
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI error string: Password and confirmation do not match.

---

## 1.5.133 BS\_GUI\_ERROR\_02 Macro

**C++**

```
#define BS_GUI_ERROR_02 777
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI error string: Password is empty.

---

## 1.5.134 BS\_GUI\_RESOURCE\_01 Macro

**C++**

```
#define BS_GUI_RESOURCE_01 730
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: CD/DVD Eraser (Title of erase dialog)

**Example**

## 1.5.135 BS\_GUI\_RESOURCE\_02 Macro

**C++**

```
#define BS_GUI_RESOURCE_02 731
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: CD/DVD Burner (Title of burn window).

**Example**

---

## 1.5.136 BS\_GUI\_RESOURCE\_03 Macro

**C++**

```
#define BS_GUI_RESOURCE_03 732
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Device

**Example**

---

## 1.5.137 BS\_GUI\_RESOURCE\_04 Macro

**C++**

```
#define BS_GUI_RESOURCE_04 733
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Progress

**Example**

---

## 1.5.138 BS\_GUI\_RESOURCE\_05 Macro

**C++**

```
#define BS_GUI_RESOURCE_05 734
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Status

**Example**

---

## 1.5.139 BS\_GUI\_RESOURCE\_06 Macro

**C++**

```
#define BS_GUI_RESOURCE_06 735
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Fast

**Example**

---

## 1.5.140 BS\_GUI\_RESOURCE\_07 Macro

**C++**

```
#define BS_GUI_RESOURCE_07 736
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Complete

**Example**

---

## 1.5.141 BS\_GUI\_RESOURCE\_08 Macro

**C++**

```
#define BS_GUI_RESOURCE_08 737
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Exit

**Example**

---

## 1.5.142 BS\_GUI\_RESOURCE\_09 Macro

**C++**

```
#define BS_GUI_RESOURCE_09 738
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Settings

**Example**

---

## 1.5.143 BS\_GUI\_RESOURCE\_10 Macro

**C++**

```
#define BS_GUI_RESOURCE_10 739
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Start

**Example**

---

## 1.5.144 BS\_GUI\_RESOURCE\_11 Macro

**C++**

```
#define BS_GUI_RESOURCE_11 740
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Stop

**Example**

---

## 1.5.145 BS\_GUI\_RESOURCE\_12 Macro

**C++**

```
#define BS_GUI_RESOURCE_12 741
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Simulate burning

**Example**

## 1.5.146 BS\_GUI\_RESOURCE\_13 Macro

**C++**

```
#define BS_GUI_RESOURCE_13 742
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Finalize medium

**Example**

---

## 1.5.147 BS\_GUI\_RESOURCE\_14 Macro

**C++**

```
#define BS_GUI_RESOURCE_14 743
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Finalize medium

**Example**

---

## 1.5.148 BS\_GUI\_RESOURCE\_15 Macro

**C++**

```
#define BS_GUI_RESOURCE_15 744
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Speed

**Example**

---

## 1.5.149 BS\_GUI\_RESOURCE\_16 Macro

**C++**

```
#define BS_GUI_RESOURCE_16 745
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Eject medium after burn

**Example**

---

## 1.5.150 BS\_GUI\_RESOURCE\_17 Macro

**C++**

```
#define BS_GUI_RESOURCE_17 746
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Joliet file system

**Example**

---

## 1.5.151 BS\_GUI\_RESOURCE\_18 Macro

**C++**

```
#define BS_GUI_RESOURCE_18 747
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Cache size

**Example**

---

## 1.5.152 BS\_GUI\_RESOURCE\_19 Macro

**C++**

```
#define BS_GUI_RESOURCE_19 748
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: 0.5 MB

**Example**

---

## 1.5.153 BS\_GUI\_RESOURCE\_20 Macro

**C++**

```
#define BS_GUI_RESOURCE_20 749
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: 64 MB

**Example**

---

## 1.5.154 BS\_GUI\_RESOURCE\_21 Macro

**C++**

```
#define BS_GUI_RESOURCE_21 750
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Medium name

**Example**

---

## 1.5.155 BS\_GUI\_RESOURCE\_22 Macro

**C++**

```
#define BS_GUI_RESOURCE_22 751
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Make boot medium

**Example**

---

## 1.5.156 BS\_GUI\_RESOURCE\_23 Macro

**C++**

```
#define BS_GUI_RESOURCE_23 752
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: No of Copies

**Example**

## 1.5.157 BS\_GUI\_RESOURCE\_24 Macro

**C++**

```
#define BS_GUI_RESOURCE_24 753
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Medium

**Example**

---

## 1.5.158 BS\_GUI\_RESOURCE\_25 Macro

**C++**

```
#define BS_GUI_RESOURCE_25 754
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Burning

**Example**

---

## 1.5.159 BS\_GUI\_RESOURCE\_26 Macro

**C++**

```
#define BS_GUI_RESOURCE_26 755
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: General

**Example**

---

## 1.5.160 BS\_GUI\_RESOURCE\_27 Macro

**C++**

```
#define BS_GUI_RESOURCE_27 756
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Simulation was successful

---

## 1.5.161 BS\_GUI\_RESOURCE\_28 Macro

**C++**

```
#define BS_GUI_RESOURCE_28 757
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Burn ([see page 121](#)) the medium now?

---

## 1.5.162 BS\_GUI\_RESOURCE\_29 Macro

**C++**

```
#define BS_GUI_RESOURCE_29 758
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Verify data after burn

**Example**

---

## 1.5.163 BS\_GUI\_RESOURCE\_30 Macro

**C++**

```
#define BS_GUI_RESOURCE_30 759
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: DVD High-Compatibility-Mode

---

---

## 1.5.164 BS\_GUI\_RESOURCE\_31 Macro

**C++**

```
#define BS_GUI_RESOURCE_31 760
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

---

**Description**

GUI resource string: Bytes Written:

**Example**

---

## 1.5.165 BS\_GUI\_RESOURCE\_32 Macro

**C++**

```
#define BS_GUI_RESOURCE_32 761
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Elapsed Time:

**Example**

---

## 1.5.166 BS\_GUI\_RESOURCE\_33 Macro

**C++**

```
#define BS_GUI_RESOURCE_33 762
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Buffer:

**Example**

---

## 1.5.167 BS\_GUI\_RESOURCE\_34 Macro

**C++**

```
#define BS_GUI_RESOURCE_34 763
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

GUI resource string: Remaining Time:

**Example**

---

## 1.5.168 BS\_GUI\_RESOURCE\_35 Macro

**C++**

```
#define BS_GUI_RESOURCE_35 764
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Other

**Example**

---

## 1.5.169 BS\_GUI\_RESOURCE\_36 Macro

**C++**

```
#define BS_GUI_RESOURCE_36 765
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Compression

**Example**

---

## 1.5.170 BS\_GUI\_RESOURCE\_37 Macro

**C++**

```
#define BS_GUI_RESOURCE_37 766
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Encryption

**Example**

---

## 1.5.171 BS\_GUI\_RESOURCE\_38 Macro

**C++**

```
#define BS_GUI_RESOURCE_38 767
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Password

**Example**

## 1.5.172 BS\_GUI\_RESOURCE\_39 Macro

**C++**

```
#define BS_GUI_RESOURCE_39 768
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: UDF

**Example**

---

## 1.5.173 BS\_GUI\_RESOURCE\_40 Macro

**C++**

```
#define BS_GUI_RESOURCE_40 769
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Version

**Example**

---

## 1.5.174 BS\_GUI\_RESOURCE\_41 Macro

**C++**

```
#define BS_GUI_RESOURCE_41 770
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Partition type

**Example**

---

## 1.5.175 BS\_GUI\_RESOURCE\_42 Macro

**C++**

```
#define BS_GUI_RESOURCE_42 771
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Save Log

**Example**

---

## 1.5.176 BS\_GUI\_RESOURCE\_43 Macro

**C++**

```
#define BS_GUI_RESOURCE_43 772
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Select multiple devices

**Example**

---

## 1.5.177 BS\_GUI\_RESOURCE\_44 Macro

**C++**

```
#define BS_GUI_RESOURCE_44 773
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Multiple devices

---

## 1.5.178 BS\_GUI\_RESOURCE\_45 Macro

**C++**

```
#define BS_GUI_RESOURCE_45 774
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Write method

**Example**

---

## 1.5.179 BS\_GUI\_RESOURCE\_46 Macro

**C++**

```
#define BS_GUI_RESOURCE_46 775
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Confirm

---

## 1.5.180 BS\_GUI\_RESOURCE\_47 Macro

**C++**

```
#define BS_GUI_RESOURCE_47 778
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Write streams

**Example**

---

## 1.5.181 BS\_GUI\_RESOURCE\_48 Macro

**C++**

```
#define BS_GUI_RESOURCE_48 779
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Rescan

**Example**

---

## 1.5.182 BS\_GUI\_RESOURCE\_49 Macro

**C++**

```
#define BS_GUI_RESOURCE_49 780
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Auto Erase (see page 153)

**Example**

## 1.5.183 BS\_GUI\_RESOURCE\_50 Macro

**C++**

```
#define BS_GUI_RESOURCE_50 781
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Rock Ridge File System

**Example**

---

## 1.5.184 BS\_GUI\_RESOURCE\_51 Macro

**C++**

```
#define BS_GUI_RESOURCE_51 782
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Pad Data Tracks

**Example**

---

## 1.5.185 BS\_GUI\_RESOURCE\_52 Macro

**C++**

```
#define BS_GUI_RESOURCE_52 783
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: Eject (Erase ([see page 153](#)) Dialog)

**Example**

---

## 1.5.186 BS\_GUI\_RESOURCE\_53 Macro

**C++**

```
#define BS_GUI_RESOURCE_53 784
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

GUI resource string: AVCHD disc

**Example**

---

**1.5.187****BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_ARTIST\_COLUMN**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLUMN 787
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Artist

---

**1.5.188****BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_CATEGORY\_COLUMN**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_COLUMN 786
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Category

---

**1.5.189****BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_DISCID\_COLUMN**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLUMN 789
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: DisclD

---

**1.5.190****BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_STATUS\_LABEL**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABEL 790
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Status

---

**1.5.191 BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_TITLE**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE 785
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Title

---

**1.5.192****BS\_GUI\_RESOURCE\_INTERNET\_DIALOG\_TITLE\_COLUMN**  
**Macro****C++**

```
#define BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN 788
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

GUI resource string: Title

---

## 1.5.193 BS\_HD\_DVD\_R Macro

**C++**

```
#define BS_HD_DVD_R 22
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

A disc identifier for disc type: HDDVD R

---

## 1.5.194 BS\_HD\_DVD\_R\_DL Macro

**C++**

```
#define BS_HD_DVD_R_DL 26
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

A disc identifier for disc type: HDDVD R Double Layer

---

## 1.5.195 BS\_HD\_DVD\_RAM Macro

**C++**

```
#define BS_HD_DVD_RAM 25
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

A disc identifier for disc type: HDDVD-RAM

---

## 1.5.196 BS\_HD\_DVD\_ROM Macro

**C++**

```
#define BS_HD_DVD_ROM 24
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

A disc identifier for disc type: HDDVD-ROM

---

## 1.5.197 BS\_HD\_DVD\_RW Macro

**C++**

```
#define BS_HD_DVD_RW 23
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

A disc identifier for disc type: HDDVD-ReWriteable

---

## 1.5.198 BS\_HD\_DVD\_RW\_DL Macro

**C++**

```
#define BS_HD_DVD_RW_DL 27
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

A disc identifier for disc type: HDDVD ReWriteable Double Layer

---

## 1.5.199 BS\_IL\_HIGH\_DEBUG Macro

**C++**

```
#define BS_IL_HIGH_DEBUG 3
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Log info level: High. Use only for debugging.

---

## 1.5.200 BS\_IL\_INFO Macro

**C++**

```
#define BS_IL_INFO 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Log info level: Info. Use this for your own log files.

## 1.5.201 BS\_IL\_LOW\_DEBUG Macro

**C++**

```
#define BS_IL_LOW_DEBUG 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Log info level: Low. Use only for debugging.

---

## 1.5.202 BS\_IL\_MEDIUM\_DEBUG Macro

**C++**

```
#define BS_IL_MEDIUM_DEBUG 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Log info level: Medium. Use only for debugging.

---

## 1.5.203 BS\_IIE\_TOO\_BIG\_FILE Macro

**C++**

```
#define BS_IIE_TOO_BIG_FILE 172
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

ISO level error: File is to Big.

---

## 1.5.204 BS\_IIE\_TOO\_LONG\_DIRECTORY\_NESTING Macro

**C++**

```
#define BS_IIE_TOO_LONG_DIRECTORY_NESTING 171
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

ISO level error: To long Directory

## 1.5.205 BS\_IMG\_BIN Macro

**C++**

```
#define BS_IMG_BIN 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A image type BIN (RAW) with a blocksize of 2352.

---

## 1.5.206 BS\_IMG\_ISO Macro

**C++**

```
#define BS_IMG_ISO 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A image type ISO with a blocksize of 2048.

---

## 1.5.207 BS\_IMGTASK\_CREATE Macro

**C++**

```
#define BS_IMGTASK_CREATE 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

The IsoSDK will create a image file.

---

## 1.5.208 BS\_IMGTASK\_VERIFY Macro

**C++**

```
#define BS_IMGTASK_VERIFY 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

The IsoSDK will verify a image file.

## 1.5.209 BS\_IMPOPTS\_COMMON Macro

**C++**

```
#define BS_IMPOPTS_COMMON 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_IMPOPTS\_COMMON.

---

## 1.5.210 BS\_IMPOPTS\_DECRYPT Macro

**C++**

```
#define BS_IMPOPTS_DECRYPT 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_IMPOPTS\_DECRYPT.

---

## 1.5.211 BS\_IMPOPTS\_EX Macro

**C++**

```
#define BS_IMPOPTS_EX 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_IMPOPTS\_EX.

---

## 1.5.212 BS\_IMPOPTS\_UNCOMPRESS Macro

**C++**

```
#define BS_IMPOPTS_UNCOMPRESS 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_IMPOPTS\_UNCOMPRESS.

## 1.5.213 BS\_INVALID\_HANDLE Macro

### C++

```
#define BS_INVALID_HANDLE -1
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: The submitted handle is not valid or not initialized.

---

## 1.5.214 BS\_INVALID\_TAG\_HANDLE Macro

### C++

```
#define BS_INVALID_TAG_HANDLE BS_INVALID_HANDLE
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: A created but not initialized Handle to receive Tags of a disc.

---

## 1.5.215 BS\_ISO\_LEVEL\_1 Macro

### C++

```
#define BS_ISO_LEVEL_1 1
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Disc file system: ISO Level 1

Specification:

1. Name format is 8.3. Available chars: only capital letters and underscore.
2. File extension length is limited to 3 chars.
3. Directory can't has extension.
4. File size limit is 2 GB.

---

## 1.5.216 BS\_ISO\_LEVEL\_2 Macro

### C++

```
#define BS_ISO_LEVEL_2 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc file system: ISO Level 2.

Specification:

1. Max name length is 31 chars.
2. File size limit is 2 GB.

---

## 1.5.217 BS\_ISO\_LEVEL\_3 Macro

**C++**

```
#define BS_ISO_LEVEL_3 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc file system: ISO Level 3.

Specification:

1. Max name length is 128 chars.
2. No file size limit.

---

## 1.5.218 BS\_ISO\_LEVEL\_ROMEO Macro

**C++**

```
#define BS_ISO_LEVEL_ROMEO 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc file system: ISO Level Romeo.

Specification:

1. Max name length is 128 chars.
2. Lower case chars are allowed in file names.

---

## 1.5.219 BS\_ISRC\_READ Macro

**C++**

```
#define BS_ISRC_READ 0xFFFF0000000CLL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read the ISRC (International Standard Recording Code) info of a disc.

---

## 1.5.220 BS\_LABELFLASH Macro

**C++**

```
#define BS_LABELFLASH 0xFFFF00000021LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write [Labelflash](#) discs.

---

## 1.5.221 BS\_LAYER\_JUMP\_RECORDING Macro

**C++**

```
#define BS_LAYER_JUMP_RECORDING 0x8000000000000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to perform Layer Jump Recording.

---

## 1.5.222 BS\_LIGHTSCRIBE Macro

**C++**

```
#define BS_LIGHTSCRIBE 0xFFFF00000022LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write [Lightscribe](#) discs.

---

## 1.5.223 BS\_LOCK\_MEDIA Macro

**C++**

```
#define BS_LOCK_MEDIA 0xFFFF000000FLL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to lock the media.

---

## 1.5.224 BS\_LOCK\_STATE Macro

**C++**

```
#define BS_LOCK_STATE 0xFFFF00000010LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able return the lock state.

---

## 1.5.225 BS\_LP\_BOOL Macro

**C++**

```
#define BS_LP_BOOL int8*
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_LP\_BOOL.

---

## 1.5.226 BS\_LS\_COMPLETE\_SESSION Macro

**C++**

```
#define BS_LS_COMPLETE_SESSION 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc session status: Complete.

---

## 1.5.227 BS\_LS\_DAMAGED\_SESSION Macro

**C++**

```
#define BS_LS_DAMAGED_SESSION 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc session status: Session is damaged.

## 1.5.228 BS\_LS\_EMPTY\_SESSION Macro

**C++**

```
#define BS_LS_EMPTY_SESSION 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc session status: Session is empty.

---

## 1.5.229 BS\_LS\_INCOMPLETE\_SESSION Macro

**C++**

```
#define BS_LS_INCOMPLETE_SESSION 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc session status: Session is incomplete.

---

## 1.5.230 BS\_MAX\_SPEED Macro

**C++**

```
#define BS_MAX_SPEED 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This represents the max available speed of the device.

---

## 1.5.231 BS\_METHOD\_2\_ADDRESSING\_FIXED\_PACKETS Macro

**C++**

```
#define BS_METHOD_2_ADDRESSING_FIXED_PACKETS 0xFFFF00000019LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device support reading fixed packet tracks on CD-R/RW media where the Addressing type is Method 2.

## 1.5.232 BS\_MODE2\_FORM1\_READ Macro

**C++**

```
#define BS_MODE2_FORM1_READ 0xFFFF00000005LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read disc that are recorded with Mode2 Form1 format.

---

## 1.5.233 BS\_MODE2\_FORM2\_READ Macro

**C++**

```
#define BS_MODE2_FORM2_READ 0xFFFF00000006LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read disc that are recorded with Mode2 Form2 format.

---

## 1.5.234 BS\_MS\_COMPLETE\_DISK Macro

**C++**

```
#define BS_MS_COMPLETE_DISK 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Disc status: Disc is complete / finalized.

---

## 1.5.235 BS\_MS\_EMPTY\_DISK Macro

**C++**

```
#define BS_MS_EMPTY_DISK 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Disc status: Disc is empty

## 1.5.236 BS\_MS\_INCOMPLETE\_DISK Macro

**C++**

```
#define BS_MS_INCOMPLETE_DISK 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc status: Disc is incomplete (not finalized).

---

## 1.5.237 BS\_MS\_OTHER Macro

**C++**

```
#define BS_MS_OTHER 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Disc status: Unknown

---

## 1.5.238 BS\_MULTI\_ERROR\_01 Macro

**C++**

```
#define BS_MULTI_ERROR_01 457
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The disc are not identical.

---

## 1.5.239 BS\_MULTI\_ERROR\_02 Macro

**C++**

```
#define BS_MULTI_ERROR_02 458
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid write speed.

## 1.5.240 BS\_MULTISESSION Macro

**C++**

```
#define BS_MULTISESSION 0x008000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to perform multi-session disc.

---

## 1.5.241 BS\_NDEF Macro

**C++**

```
#define BS_NDEF -1 // -> BS_NDEF is interpreted as BS_TRUE.
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_NDEF.

---

## 1.5.242 BS\_PACKET\_WRITE Macro

**C++**

```
#define BS_PACKET_WRITE 0xFFFF00000020LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The drive can perform packet writing.

---

## 1.5.243 BS\_PARENTDIR\_ONLY Macro

**C++**

```
#define BS_PARENTDIR_ONLY 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Only the superordinate directory of the file will be created in chDestinationPath of the project. Example: chSourceDir = c:\tmp1\tmp2\test.tmp - only the subfolder tmp2 will be created.

## 1.5.244 BS\_PREVENT\_JUMPER Macro

### C++

```
#define BS_PREVENT_JUMPER 0xFFFF00000011LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit has a physical jumper named the Prevent/Allow Jumper and the jumper is present.

---

## 1.5.245 BS\_R\_W\_SUBCHANNELS\_DEINT\_AND\_CORR Macro

### C++

```
#define BS_R_W_SUBCHANNELS_DEINT_AND_CORR 0xFFFF0000000ALL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command with the returned data de-interleaved and error corrected

---

## 1.5.246 BS\_R\_W\_SUBCHANNELS\_IN\_LEAD\_IN\_READ Macro

### C++

```
#define BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ 0xFFFF00000018LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: The Logical Unit is capable of reading the raw R-W Sub-channel information from the Lead-in.

---

## 1.5.247 BS\_R\_W\_SUBCHANNELS\_READ Macro

### C++

```
#define BS_R_W_SUBCHANNELS_READ 0xFFFF00000009LL
```

### File

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command.

---

## 1.5.248 BS\_RDT\_DATA Macro

**C++**

```
#define BS_RDT_DATA 0x0004      // depends on track type
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Input data contains UserData field

---

## 1.5.249 BS\_RDT\_EDC\_ECC Macro

**C++**

```
#define BS_RDT_EDC_ECC 0x0008      // 0/288/0/280/4 bytes depend on track type
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Input data contains ECC (Error Correction Code) /EDC (Error Detection Code) fields

---

## 1.5.250 BS\_RDT\_SUBCH\_PQ Macro

**C++**

```
#define BS_RDT_SUBCH_PQ 0x0010      // 16 bytes
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Input data contains P-Q subchannel data in packed form

---

## 1.5.251 BS\_RDT\_SUBCH\_PW Macro

**C++**

```
#define BS_RDT_SUBCH_PW 0x0020      // 96 bytes
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Input data contains P-W subchannel data in raw form.

---

## 1.5.252 BS\_RDT\_SUBCH\_RW Macro

**C++**

```
#define BS_RDT_SUBCH_RW 0x0040 // 96 bytes
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Input data contains R-W subchannel data in raw form.

---

## 1.5.253 BS\_RDT\_SUBHEADERS Macro

**C++**

```
#define BS_RDT_SUBHEADERS 0x0002 // 8 bytes
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Input data contains Subheaders field.

---

## 1.5.254 BS\_RDT\_SYNC\_HEADER Macro

**C++**

```
#define BS_RDT_SYNC_HEADER 0x0001 // 16 bytes
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Input data contains Sync/Header field.

---

## 1.5.255 BS\_READ\_BLURAY\_R Macro

**C++**

```
#define BS_READ_BLURAY_R 0x00000000400LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read Blu-Ray R.

## 1.5.256 BS\_READ\_BLURAY\_R\_XL Macro

**C++**

```
#define BS_READ_BLURAY_R_XL 0xFFFF00000023LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read BD-R XL.

---

## 1.5.257 BS\_READ\_BLURAY\_RE Macro

**C++**

```
#define BS_READ_BLURAY_RE 0x000000000800LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read Blu-Ray ReWriteable.

---

## 1.5.258 BS\_READ\_BLURAY\_RE\_XL Macro

**C++**

```
#define BS_READ_BLURAY_RE_XL 0xFFFF00000024LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read BD-RE XL (Re-Writeable).

---

## 1.5.259 BS\_READ\_BLURAY\_ROM Macro

**C++**

```
#define BS_READ_BLURAY_ROM 0x000000001000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read Blu-Ray ROM.

## 1.5.260 BS\_READ\_CDR Macro

**C++**

```
#define BS_READ_CDR 0x000000000001LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read CD-R.

---

## 1.5.261 BS\_READ\_CDRW Macro

**C++**

```
#define BS_READ_CDRW 0x000000000002LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read CD ReWritable.

---

## 1.5.262 BS\_READ\_CDRW\_CAV Macro

**C++**

```
#define BS_READ_CDRW_CAV 0x000000040000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read CD-RW media that is designed for CAV recording.

---

## 1.5.263 BS\_READ\_DEVICE Macro

**C++**

```
#define BS_READ_DEVICE (-2)
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK device type: Read Device

## 1.5.264 BS\_READ\_DVD Macro

**C++**

```
#define BS_READ_DVD 0x000000000004LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD-R.

---

## 1.5.265 BS\_READ\_DVD\_DL Macro

**C++**

```
#define BS_READ_DVD_DL 0x000000000100LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD-R Double Layer.

---

## 1.5.266 BS\_READ\_DVD\_MRDL Macro

**C++**

```
#define BS_READ_DVD_MRDL 0x000000000200LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD-R Double Layer.

---

## 1.5.267 BS\_READ\_DVD\_RDL\_PLUS Macro

**C++**

```
#define BS_READ_DVD_RDL_PLUS 0x080000000000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD+R Double Layer.

## 1.5.268 BS\_READ\_DVD\_RWDL\_PLUS Macro

**C++**

```
#define BS_READ_DVD_RWDL_PLUS 0x1000000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read DVD+ReWritable Double Layer.

---

## 1.5.269 BS\_READ\_DVDR Macro

**C++**

```
#define BS_READ_DVDR 0x000000000008LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read DVD-R.

---

## 1.5.270 BS\_READ\_DVDR\_PLUS Macro

**C++**

```
#define BS_READ_DVDR_PLUS 0x0000000000040LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read DVD+R.

---

## 1.5.271 BS\_READ\_DVDRAM Macro

**C++**

```
#define BS_READ_DVDRAM 0x000000000020LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to read DVD-RAM.

## 1.5.272 BS\_READ\_DVDRW Macro

**C++**

```
#define BS_READ_DVDRW 0x0000000000010LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD-ReWritable.

---

## 1.5.273 BS\_READ\_DVDRW\_PLUS Macro

**C++**

```
#define BS_READ_DVDRW_PLUS 0x0000000000080LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read DVD+ReWritable.

---

## 1.5.274 BS\_READ\_HDDVD\_R Macro

**C++**

```
#define BS_READ_HDDVD_R 0x000000002000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read HDDVD R.

---

## 1.5.275 BS\_READ\_HDDVD\_ROM Macro

**C++**

```
#define BS_READ_HDDVD_ROM 0x000000008000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read HDDVD Rom.

---

## 1.5.276 BS\_READ\_HDDVD\_RW Macro

**C++**

```
#define BS_READ_HDDVD_RW 0x000000004000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read HDDVD ReWritable.

---

## 1.5.277 BS\_READ\_MOUNT\_RAINER Macro

**C++**

```
#define BS_READ_MOUNT_RAINER 0x000000020000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read Mount Rainer.

---

## 1.5.278 BS\_RM\_RAW Macro

**C++**

```
#define BS_RM_RAW 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK read mode: RAW (2352)

---

## 1.5.279 BS\_RM\_RAW\_SUBCHANNEL Macro

**C++**

```
#define BS_RM_RAW_SUBCHANNEL 2
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK read mode: RAW Subchannel (2352 +16)

---

## 1.5.280 BS\_RM\_USERDATA Macro

**C++**

```
#define BS_RM_USERDATA 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK read mode: Userdata

---

## 1.5.281 BS\_RTF\_AUDIO Macro

**C++**

```
#define BS_RTF_AUDIO 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Track type: Audio track

---

## 1.5.282 BS\_RTF\_MODE1 Macro

**C++**

```
#define BS_RTF_MODE1 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Track type: Mode 1 track

---

## 1.5.283 BS\_RTF\_MODE2\_FORM1 Macro

**C++**

```
#define BS_RTF_MODE2_FORM1 3
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Track type: [Mode 2 Form 1](#) track

---

## 1.5.284 BS\_RTF\_MODE2\_FORM2 Macro

**C++**

```
#define BS_RTF_MODE2_FORM2 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Track type: [Mode 2 Form 2](#) track

---

## 1.5.285 BS\_RTF\_MODE2\_FORMLESS Macro

**C++**

```
#define BS_RTF_MODE2_FORMLESS 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Track type: [Mode 2 formless](#) track

---

## 1.5.286 BS\_SCSI\_ERROR\_001 Macro

**C++**

```
#define BS_SCSI_ERROR_001 207
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Operation in progress.

---

## 1.5.287 BS\_SCSI\_ERROR\_002 Macro

**C++**

```
#define BS_SCSI_ERROR_002 208
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Cleaning requested.

## 1.5.288 BS\_SCSI\_ERROR\_004 Macro

**C++**

```
#define BS_SCSI_ERROR_004 219
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: No reference position found

---

## 1.5.289 BS\_SCSI\_ERROR\_005 Macro

**C++**

```
#define BS_SCSI_ERROR_005 220
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Multiple peripheral devices selected.

---

## 1.5.290 BS\_SCSI\_ERROR\_006 Macro

**C++**

```
#define BS_SCSI_ERROR_006 225
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Unreachable copy target.

---

## 1.5.291 BS\_SCSI\_ERROR\_007 Macro

**C++**

```
#define BS_SCSI_ERROR_007 226
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Track following error.

## 1.5.292 BS\_SCSI\_ERROR\_008 Macro

**C++**

```
#define BS_SCSI_ERROR_008 227
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Tracking servo failure.

---

## 1.5.293 BS\_SCSI\_ERROR\_009 Macro

**C++**

```
#define BS_SCSI_ERROR_009 228
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Focus servo failure

---

## 1.5.294 BS\_SCSI\_ERROR\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_01 200
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: No additional sense information

---

## 1.5.295 BS\_SCSI\_ERROR\_010 Macro

**C++**

```
#define BS_SCSI_ERROR_010 229
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Spindle servo failure.

## 1.5.296 BS\_SCSI\_ERROR\_011 Macro

**C++**

```
#define BS_SCSI_ERROR_011 230
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Head select fault.

---

## 1.5.297 BS\_SCSI\_ERROR\_012 Macro

**C++**

```
#define BS_SCSI_ERROR_012 248
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Head select fault.

---

## 1.5.298 BS\_SCSI\_ERROR\_013 Macro

**C++**

```
#define BS_SCSI_ERROR_013 249
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: I-ec uncorrectable error.

---

## 1.5.299 BS\_SCSI\_ERROR\_014 Macro

**C++**

```
#define BS_SCSI_ERROR_014 255
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Read error - Loss of streaming.

## 1.5.300 BS\_SCSI\_ERROR\_015 Macro

**C++**

```
#define BS_SCSI_ERROR_015 256
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Recorded entity not found.

---

## 1.5.301 BS\_SCSI\_ERROR\_016 Macro

**C++**

```
#define BS_SCSI_ERROR_016 257
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Record not found.

---

## 1.5.302 BS\_SCSI\_ERROR\_017 Macro

**C++**

```
#define BS_SCSI_ERROR_017 258
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Random positioning error.

---

## 1.5.303 BS\_SCSI\_ERROR\_018 Macro

**C++**

```
#define BS_SCSI_ERROR_018 278
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Parameter list length error.

## 1.5.304 BS\_SCSI\_ERROR\_019 Macro

**C++**

```
#define BS_SCSI_ERROR_019 279
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Synchronous data transfer error.

---

## 1.5.305 BS\_SCSI\_ERROR\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_02 201
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: I/O process terminated.

---

## 1.5.306 BS\_SCSI\_ERROR\_020 Macro

**C++**

```
#define BS_SCSI_ERROR_020 280
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Miss compare during verify operation.

---

## 1.5.307 BS\_SCSI\_ERROR\_021 Macro

**C++**

```
#define BS_SCSI_ERROR_021 281
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid command operation code.

## 1.5.308 BS\_SCSI\_ERROR\_022 Macro

**C++**

```
#define BS_SCSI_ERROR_022 282
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Logical block address out of range.

---

## 1.5.309 BS\_SCSI\_ERROR\_023 Macro

**C++**

```
#define BS_SCSI_ERROR_023 283
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid element address.

---

## 1.5.310 BS\_SCSI\_ERROR\_024 Macro

**C++**

```
#define BS_SCSI_ERROR_024 284
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid address for write.

---

## 1.5.311 BS\_SCSI\_ERROR\_025 Macro

**C++**

```
#define BS_SCSI_ERROR_025 292
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid release of persistent reservation.

## 1.5.312 BS\_SCSI\_ERROR\_026 Macro

**C++**

```
#define BS_SCSI_ERROR_026 293
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Data decryption error.

---

## 1.5.313 BS\_SCSI\_ERROR\_027 Macro

**C++**

```
#define BS_SCSI_ERROR_027 299
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Inline data length exceeded.

---

## 1.5.314 BS\_SCSI\_ERROR\_028 Macro

**C++**

```
#define BS_SCSI_ERROR_028 300
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Invalid operation for copy source or destination.

---

## 1.5.315 BS\_SCSI\_ERROR\_029 Macro

**C++**

```
#define BS_SCSI_ERROR_029 309
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Not ready to ready change, medium may have changed.

## 1.5.316 BS\_SCSI\_ERROR\_030 Macro

**C++**

```
#define BS_SCSI_ERROR_030 310
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Import or export element accessed.

---

## 1.5.317 BS\_SCSI\_ERROR\_031 Macro

**C++**

```
#define BS_SCSI_ERROR_031 311
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Power on, reset, or bus device reset occurred.

---

## 1.5.318 BS\_SCSI\_ERROR\_032 Macro

**C++**

```
#define BS_SCSI_ERROR_032 312
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Power on occurred.

---

## 1.5.319 BS\_SCSI\_ERROR\_033 Macro

**C++**

```
#define BS_SCSI_ERROR_033 313
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: SCSI bus reset occurred.

## 1.5.320 BS\_SCSI\_ERROR\_034 Macro

**C++**

```
#define BS_SCSI_ERROR_034 314
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Bus device reset function occurred.

---

## 1.5.321 BS\_SCSI\_ERROR\_035 Macro

**C++**

```
#define BS_SCSI_ERROR_035 315
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Device internal reset.

---

## 1.5.322 BS\_SCSI\_ERROR\_036 Macro

**C++**

```
#define BS_SCSI_ERROR_036 316
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Transceiver mode changed to single-ended.

---

## 1.5.323 BS\_SCSI\_ERROR\_037 Macro

**C++**

```
#define BS_SCSI_ERROR_037 317
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Transceiver mode changed to LVD.

## 1.5.324 BS\_SCSI\_ERROR\_038 Macro

**C++**

```
#define BS_SCSI_ERROR_038 321
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Reservations pre-empted.

---

## 1.5.325 BS\_SCSI\_ERROR\_039 Macro

**C++**

```
#define BS_SCSI_ERROR_039 322
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Reservations released.

---

## 1.5.326 BS\_SCSI\_ERROR\_040 Macro

**C++**

```
#define BS_SCSI_ERROR_040 323
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Registrations pre-empted.

---

## 1.5.327 BS\_SCSI\_ERROR\_041 Macro

**C++**

```
#define BS_SCSI_ERROR_041 324
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Copy cannot execute since host cannot disconnect.

## 1.5.328 BS\_SCSI\_ERROR\_042 Macro

**C++**

```
#define BS_SCSI_ERROR_042 326
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Current program area is not empty.

---

## 1.5.329 BS\_SCSI\_ERROR\_043 Macro

**C++**

```
#define BS_SCSI_ERROR_043 327
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Current program area is empty.

---

## 1.5.330 BS\_SCSI\_ERROR\_044 Macro

**C++**

```
#define BS_SCSI_ERROR_044 328
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Persistent prevent conflict.

---

## 1.5.331 BS\_SCSI\_ERROR\_045 Macro

**C++**

```
#define BS_SCSI_ERROR_045 329
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Insufficient time for operation.

## 1.5.332 BS\_SCSI\_ERROR\_046 Macro

**C++**

```
#define BS_SCSI_ERROR_046 344
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Zoned formatting failed due to spare linking.

---

## 1.5.333 BS\_SCSI\_ERROR\_047 Macro

**C++**

```
#define BS_SCSI_ERROR_047 345
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure failure.

---

## 1.5.334 BS\_SCSI\_ERROR\_048 Macro

**C++**

```
#define BS_SCSI_ERROR_048 346
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure services failure.

---

## 1.5.335 BS\_SCSI\_ERROR\_049 Macro

**C++**

```
#define BS_SCSI_ERROR_049 347
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Unsupported enclosure function.

## 1.5.336 BS\_SCSI\_ERROR\_050 Macro

**C++**

```
#define BS_SCSI_ERROR_050 348
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure services unavailable.

---

## 1.5.337 BS\_SCSI\_ERROR\_051 Macro

**C++**

```
#define BS_SCSI_ERROR_051 349
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure services transfer failure.

---

## 1.5.338 BS\_SCSI\_ERROR\_052 Macro

**C++**

```
#define BS_SCSI_ERROR_052 350
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure services transfer refused.

---

## 1.5.339 BS\_SCSI\_ERROR\_053 Macro

**C++**

```
#define BS_SCSI_ERROR_053 367
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid bits in identify message.

## 1.5.340 BS\_SCSI\_ERROR\_054 Macro

**C++**

```
#define BS_SCSI_ERROR_054 373
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Target operating conditions have changed.

---

## 1.5.341 BS\_SCSI\_ERROR\_055 Macro

**C++**

```
#define BS_SCSI_ERROR_055 374
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Microcode has been changed.

---

## 1.5.342 BS\_SCSI\_ERROR\_056 Macro

**C++**

```
#define BS_SCSI_ERROR_056 375
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Changed operating definition.

---

## 1.5.343 BS\_SCSI\_ERROR\_057 Macro

**C++**

```
#define BS_SCSI_ERROR_057 376
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Inquiry data has changed.

## 1.5.344 BS\_SCSI\_ERROR\_058 Macro

**C++**

```
#define BS_SCSI_ERROR_058 377
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Component device attached.

---

## 1.5.345 BS\_SCSI\_ERROR\_059 Macro

**C++**

```
#define BS_SCSI_ERROR_059 378
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Device identifier change.

---

## 1.5.346 BS\_SCSI\_ERROR\_060 Macro

**C++**

```
#define BS_SCSI_ERROR_060 379
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Redundancy group created or modified.

---

## 1.5.347 BS\_SCSI\_ERROR\_061 Macro

**C++**

```
#define BS_SCSI_ERROR_061 380
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Redundancy group deleted.

## 1.5.348 BS\_SCSI\_ERROR\_062 Macro

**C++**

```
#define BS_SCSI_ERROR_062 381
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Spare created or modified.

---

## 1.5.349 BS\_SCSI\_ERROR\_063 Macro

**C++**

```
#define BS_SCSI_ERROR_063 382
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Spare deleted.

---

## 1.5.350 BS\_SCSI\_ERROR\_064 Macro

**C++**

```
#define BS_SCSI_ERROR_064 387
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Reported luns data has changed.

---

## 1.5.351 BS\_SCSI\_ERROR\_065 Macro

**C++**

```
#define BS_SCSI_ERROR_065 388
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Echo buffer overwritten.

## 1.5.352 BS\_SCSI\_ERROR\_066 Macro

**C++**

```
#define BS_SCSI_ERROR_066 391
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Message error.

---

## 1.5.353 BS\_SCSI\_ERROR\_067 Macro

**C++**

```
#define BS_SCSI_ERROR_067 392
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Internal target failure.

---

## 1.5.354 BS\_SCSI\_ERROR\_068 Macro

**C++**

```
#define BS_SCSI_ERROR_068 393
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Select or reselect failure.

---

## 1.5.355 BS\_SCSI\_ERROR\_069 Macro

**C++**

```
#define BS_SCSI_ERROR_069 394
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Unsuccessful soft reset.

## 1.5.356 BS\_SCSI\_ERROR\_070 Macro

**C++**

```
#define BS_SCSI_ERROR_070 395
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: SCSI parity error.

---

## 1.5.357 BS\_SCSI\_ERROR\_071 Macro

**C++**

```
#define BS_SCSI_ERROR_071 396
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Data phase crc error detected.

---

## 1.5.358 BS\_SCSI\_ERROR\_072 Macro

**C++**

```
#define BS_SCSI_ERROR_072 397
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: SCSI parity error detected during st data phase.

---

## 1.5.359 BS\_SCSI\_ERROR\_073 Macro

**C++**

```
#define BS_SCSI_ERROR_073 398
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Information unit crc error detected.

## 1.5.360 BS\_SCSI\_ERROR\_074 Macro

**C++**

```
#define BS_SCSI_ERROR_074 399
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Asynchronous information protection error detected.

---

## 1.5.361 BS\_SCSI\_ERROR\_075 Macro

**C++**

```
#define BS_SCSI_ERROR_075 400
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Initiator detected error message received.

---

## 1.5.362 BS\_SCSI\_ERROR\_076 Macro

**C++**

```
#define BS_SCSI_ERROR_076 401
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid message error.

---

## 1.5.363 BS\_SCSI\_ERROR\_077 Macro

**C++**

```
#define BS_SCSI_ERROR_077 403
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Data phase error.

## 1.5.364 BS\_SCSI\_ERROR\_078 Macro

**C++**

```
#define BS_SCSI_ERROR_078 410
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Insufficient reservation resources.

---

## 1.5.365 BS\_SCSI\_ERROR\_079 Macro

**C++**

```
#define BS_SCSI_ERROR_079 411
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Insufficient resources.

---

## 1.5.366 BS\_SCSI\_ERROR\_080 Macro

**C++**

```
#define BS_SCSI_ERROR_080 412
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Insufficient registration resources.

---

## 1.5.367 BS\_SCSI\_ERROR\_081 Macro

**C++**

```
#define BS_SCSI_ERROR_081 413
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Unable to recover table-of-contents.

## 1.5.368 BS\_SCSI\_ERROR\_082 Macro

**C++**

```
#define BS_SCSI_ERROR_082 414
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Operator request or state change input.

---

## 1.5.369 BS\_SCSI\_ERROR\_083 Macro

**C++**

```
#define BS_SCSI_ERROR_083 415
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Operator medium removal request.

---

## 1.5.370 BS\_SCSI\_ERROR\_084 Macro

**C++**

```
#define BS_SCSI_ERROR_084 416
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Operator selected write protect.

---

## 1.5.371 BS\_SCSI\_ERROR\_085 Macro

**C++**

```
#define BS_SCSI_ERROR_085 417
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Operator selected write permit.

## 1.5.372 BS\_SCSI\_ERROR\_086 Macro

**C++**

```
#define BS_SCSI_ERROR_086 419
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Threshold condition met.

---

## 1.5.373 BS\_SCSI\_ERROR\_087 Macro

**C++**

```
#define BS_SCSI_ERROR_087 422
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Failure prediction threshold exceeded.

---

## 1.5.374 BS\_SCSI\_ERROR\_088 Macro

**C++**

```
#define BS_SCSI_ERROR_088 425
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Spare area exhaustion prediction threshold exceeded.

---

## 1.5.375 BS\_SCSI\_ERROR\_089 Macro

**C++**

```
#define BS_SCSI_ERROR_089 426
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Failure prediction threshold exceeded (false).

## 1.5.376 BS\_SCSI\_ERROR\_090 Macro

**C++**

```
#define BS_SCSI_ERROR_090 427
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Low power condition on.

---

## 1.5.377 BS\_SCSI\_ERROR\_091 Macro

**C++**

```
#define BS_SCSI_ERROR_091 428
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Idle condition activated by timer.

---

## 1.5.378 BS\_SCSI\_ERROR\_092 Macro

**C++**

```
#define BS_SCSI_ERROR_092 429
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Standby condition activated by timer.

---

## 1.5.379 BS\_SCSI\_ERROR\_093 Macro

**C++**

```
#define BS_SCSI_ERROR_093 430
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Idle condition activated by command.

## 1.5.380 BS\_SCSI\_ERROR\_094 Macro

**C++**

```
#define BS_SCSI_ERROR_094 431
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Standby condition activated by command.

---

## 1.5.381 BS\_SCSI\_ERROR\_095 Macro

**C++**

```
#define BS_SCSI_ERROR_095 432
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: End of user area encountered on this track.

---

## 1.5.382 BS\_SCSI\_ERROR\_096 Macro

**C++**

```
#define BS_SCSI_ERROR_096 433
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Packet does not fit in available space.

---

## 1.5.383 BS\_SCSI\_ERROR\_097 Macro

**C++**

```
#define BS_SCSI_ERROR_097 434
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Illegal mode for this track.

## 1.5.384 BS\_SCSI\_ERROR\_098 Macro

**C++**

```
#define BS_SCSI_ERROR_098 435
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid packet size.

---

## 1.5.385 BS\_SCSI\_ERROR\_099 Macro

**C++**

```
#define BS_SCSI_ERROR_099 436
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Voltage fault.

---

## 1.5.386 BS\_SCSI\_ERROR\_100 Macro

**C++**

```
#define BS_SCSI_ERROR_100 447
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Empty or partially written reserved track.

---

## 1.5.387 BS\_SCSI\_ERROR\_101 Macro

**C++**

```
#define BS_SCSI_ERROR_101 448
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: No more track reservations allowed.

## 1.5.388 BS\_SCSI\_ERROR\_102 Macro

**C++**

```
#define BS_SCSI_ERROR_102 449
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: CD control error.

---

## 1.5.389 BS\_SCSI\_ERROR\_103 Macro

**C++**

```
#define BS_SCSI_ERROR_103 450
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Power calibration area almost full.

---

## 1.5.390 BS\_SCSI\_ERROR\_104 Macro

**C++**

```
#define BS_SCSI_ERROR_104 451
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Power calibration area is full.

---

## 1.5.391 BS\_SCSI\_ERROR\_105 Macro

**C++**

```
#define BS_SCSI_ERROR_105 452
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Power calibration area error.

## 1.5.392 BS\_SCSI\_ERROR\_106 Macro

**C++**

```
#define BS_SCSI_ERROR_106 455
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: RMA/PMA is almost full.

---

## 1.5.393 BS\_SCSI\_ERROR\_107 Macro

**C++**

```
#define BS_SCSI_ERROR_107 113
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Failed to request sense.

---

## 1.5.394 BS\_SCSI\_ERROR\_108 Macro

**C++**

```
#define BS_SCSI_ERROR_108 460
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium load or eject failed.

---

## 1.5.395 BS\_SCSI\_ERROR\_109 Macro

**C++**

```
#define BS_SCSI_ERROR_109 461
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium removal prevented.

## 1.5.396 BS\_SCSI\_ERROR\_110 Macro

**C++**

```
#define BS_SCSI_ERROR_110 464
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid write crossing layer jump.

---

## 1.5.397 BS\_SCSI\_ERROR\_111 Macro

**C++**

```
#define BS_SCSI_ERROR_111 465
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Illegal function.

---

## 1.5.398 BS\_SCSI\_ERROR\_112 Macro

**C++**

```
#define BS_SCSI_ERROR_112 466
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Format-layer may have changed.

---

## 1.5.399 BS\_SCSI\_ERROR\_113 Macro

**C++**

```
#define BS_SCSI_ERROR_113 468
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Enclosure services checksum error.

## 1.5.400 BS\_SCSI\_ERROR\_114 Macro

**C++**

```
#define BS_SCSI_ERROR_114 469
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: System resource failure.

---

## 1.5.401 BS\_SCSI\_ERROR\_115 Macro

**C++**

```
#define BS_SCSI_ERROR_115 470
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: RMZ extention is not allowed.

---

## 1.5.402 BS\_SCSI\_ERROR\_116 Macro

**C++**

```
#define BS_SCSI_ERROR_116 471
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: No more test zone extensions are allowed.

---

## 1.5.403 BS\_SCSI\_ERROR\_117 Macro

**C++**

```
#define BS_SCSI_ERROR_117 472
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Current power calibration area is almost full.

## 1.5.404 BS\_SCSI\_ERROR\_118 Macro

**C++**

```
#define BS_SCSI_ERROR_118 473
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Current power calibration area is full.

---

## 1.5.405 BS\_SCSI\_ERROR\_119 Macro

**C++**

```
#define BS_SCSI_ERROR_119 474
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: RDZ is full.

---

## 1.5.406 BS\_SCSI\_ERROR\_ALLOC\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_ALLOC_01 453
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Program memory area update failure.

---

## 1.5.407 BS\_SCSI\_ERROR\_ALLOC\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_ALLOC_02 454
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Program memory area is full.

## 1.5.408 BS\_SCSI\_ERROR\_ASPI\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_01 5
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: An ASPI error occurred.

---

## 1.5.409 BS\_SCSI\_ERROR\_ASPI\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_02 102
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Invalid host adapter.

---

## 1.5.410 BS\_SCSI\_ERROR\_ASPI\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_03 103
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: No device has been specified.

---

## 1.5.411 BS\_SCSI\_ERROR\_ASPI\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_04 106
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: ASPI is busy.

## 1.5.412 BS\_SCSI\_ERROR\_ASPI\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_05 111
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Bus reset.

---

## 1.5.413 BS\_SCSI\_ERROR\_ASPI\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_06 6
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid device ID or device is not a CD/DVD or BD drive.

---

## 1.5.414 BS\_SCSI\_ERROR\_ASPI\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_07 141
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The device is busy.

---

## 1.5.415 BS\_SCSI\_ERROR\_ASPI\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_08 118
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The target is busy.

## 1.5.416 BS\_SCSI\_ERROR\_ASPI\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_09 119
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Target reservation conflict.

---

## 1.5.417 BS\_SCSI\_ERROR\_ASPI\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_ASPI_10 120
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The target queue is full.

---

## 1.5.418 BS\_SCSI\_ERROR\_ATT\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_ATT_01 232
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Warning.

---

## 1.5.419 BS\_SCSI\_ERROR\_ATT\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_ATT_02 233
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Warning - Specified temperature exceeded.

## 1.5.420 BS\_SCSI\_ERROR\_ATT\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_ATT_03 234
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Warning - Enclosure degraded.

---

## 1.5.421 BS\_SCSI\_ERROR\_ATT\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_ATT_04 462
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Warning - background self-test failed.

---

## 1.5.422 BS\_SCSI\_ERROR\_ATT\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_ATT_05 463
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Warning – background pre-scan detected medium error.

---

## 1.5.423 BS\_SCSI\_ERROR\_AUDIO\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_AUDIO_01 202
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Audio play operation in progress.

## 1.5.424 BS\_SCSI\_ERROR\_AUDIO\_02 Macro

### C++

```
#define BS_SCSI_ERROR_AUDIO_02 203
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Audio play operation paused.

---

## 1.5.425 BS\_SCSI\_ERROR\_AUDIO\_03 Macro

### C++

```
#define BS_SCSI_ERROR_AUDIO_03 204
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Audio play operation successfully completed.

---

## 1.5.426 BS\_SCSI\_ERROR\_AUDIO\_04 Macro

### C++

```
#define BS_SCSI_ERROR_AUDIO_04 205
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Audio play operation stopped due to error.

---

## 1.5.427 BS\_SCSI\_ERROR\_AUDIO\_05 Macro

### C++

```
#define BS_SCSI_ERROR_AUDIO_05 206
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: No current audio status to return.

## 1.5.428 BS\_SCSI\_ERROR\_CDB\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_CDB_01 285
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid field in cdb.

---

## 1.5.429 BS\_SCSI\_ERROR\_CDB\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_CDB_02 286
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: CDB decryption error.

---

## 1.5.430 BS\_SCSI\_ERROR\_CIRC\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_CIRC_01 250
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Circ unrecovered error.

---

## 1.5.431 BS\_SCSI\_ERROR\_COMMAND\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_COMMAND_02 325
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Command sequence error.

## 1.5.432 BS\_SCSI\_ERROR\_COMMAND\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_COMMAND_03 330
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Commands cleared by another initiator.

---

## 1.5.433 BS\_SCSI\_ERROR\_COMMAND\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_COMMAND_04 402
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Command phase error.

---

## 1.5.434 BS\_SCSI\_ERROR\_COMMAND\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_COMMAND_05 405
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Overlapped commands attempted.

---

## 1.5.435 BS\_SCSI\_ERROR\_CRC\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_CRC_01 251
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: De-compression CRC error.

## 1.5.436 BS\_SCSI\_ERROR\_DCSS\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_01 437
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Copy protection key exchange failure - authentication failure.

---

## 1.5.437 BS\_SCSI\_ERROR\_DCSS\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_02 438
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Copy protection key exchange failure - key not present.

---

## 1.5.438 BS\_SCSI\_ERROR\_DCSS\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_03 439
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Copy protection key exchange failure - key not established.

---

## 1.5.439 BS\_SCSI\_ERROR\_DCSS\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_04 440
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Read of scrambled sector without authentication.

## 1.5.440 BS\_SCSI\_ERROR\_DCSS\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_05 441
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium region code is mismatched to logical unit region.

---

## 1.5.441 BS\_SCSI\_ERROR\_DCSS\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_06 442
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Drive region must be permanent/region reset count error.

---

## 1.5.442 BS\_SCSI\_ERROR\_DCSS\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_07 475
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Insufficient block count for binding nonce recording.

---

## 1.5.443 BS\_SCSI\_ERROR\_DCSS\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_DCSS_08 476
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Conflict in binding nonce recording.

## 1.5.444 BS\_SCSI\_ERROR\_DECOM\_01 Macro

### C++

```
#define BS_SCSI_ERROR_DECOM_01 252
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: Cannot decompress using declared algorithm.

---

## 1.5.445 BS\_SCSI\_ERROR\_DISK\_01 Macro

### C++

```
#define BS_SCSI_ERROR_DISK_01 331
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: Incompatible medium installeds.

---

## 1.5.446 BS\_SCSI\_ERROR\_DISK\_02 Macro

### C++

```
#define BS_SCSI_ERROR_DISK_02 332
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: Cannot read medium - unknown format.

---

## 1.5.447 BS\_SCSI\_ERROR\_DISK\_03 Macro

### C++

```
#define BS_SCSI_ERROR_DISK_03 333
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK error code: Cannot read medium - incompatible format.

## 1.5.448 BS\_SCSI\_ERROR\_DISK\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_04 334
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cleaning cartridge installed.

---

## 1.5.449 BS\_SCSI\_ERROR\_DISK\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_05 335
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cannot write medium - unknown format.

---

## 1.5.450 BS\_SCSI\_ERROR\_DISK\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_06 336
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cannot write medium - incompatible format.

---

## 1.5.451 BS\_SCSI\_ERROR\_DISK\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_07 337
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cannot format medium - incompatible medium.

## 1.5.452 BS\_SCSI\_ERROR\_DISK\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_08 338
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cleaning failure.

---

## 1.5.453 BS\_SCSI\_ERROR\_DISK\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_09 339
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cannot write - application code mismatch.

---

## 1.5.454 BS\_SCSI\_ERROR\_DISK\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_10 340
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Current session not fixated for append.

---

## 1.5.455 BS\_SCSI\_ERROR\_DISK\_11 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_11 341
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not formatted.

## 1.5.456 BS\_SCSI\_ERROR\_DISK\_12 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_12 342
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium format corrupted.

---

## 1.5.457 BS\_SCSI\_ERROR\_DISK\_13 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_13 343
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Format command failed.

---

## 1.5.458 BS\_SCSI\_ERROR\_DISK\_14 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_14 353
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not present.

---

## 1.5.459 BS\_SCSI\_ERROR\_DISK\_15 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_15 354
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not present - tray closed.

## 1.5.460 BS\_SCSI\_ERROR\_DISK\_16 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_16 355
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not present - tray open.

---

## 1.5.461 BS\_SCSI\_ERROR\_DISK\_17 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_17 356
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not present - loadable.

---

## 1.5.462 BS\_SCSI\_ERROR\_DISK\_18 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_18 357
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium not present - medium auxiliary memory accessible.

---

## 1.5.463 BS\_SCSI\_ERROR\_DISK\_19 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_19 358
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium destination element full.

## 1.5.464 BS\_SCSI\_ERROR\_DISK\_20 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_20 359
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium source element empty.

---

## 1.5.465 BS\_SCSI\_ERROR\_DISK\_21 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_21 360
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: End of medium reached.

---

## 1.5.466 BS\_SCSI\_ERROR\_DISK\_22 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_22 361
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium magazine not accessible.

---

## 1.5.467 BS\_SCSI\_ERROR\_DISK\_23 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_23 362
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium magazine removed.

## 1.5.468 BS\_SCSI\_ERROR\_DISK\_24 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_24 363
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium magazine inserted.

---

## 1.5.469 BS\_SCSI\_ERROR\_DISK\_25 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_25 364
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium magazine locked.

---

## 1.5.470 BS\_SCSI\_ERROR\_DISK\_26 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_26 365
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium magazine unlocked.

---

## 1.5.471 BS\_SCSI\_ERROR\_DISK\_27 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_27 389
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium loadable.

## 1.5.472 BS\_SCSI\_ERROR\_DISK\_28 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_28 390
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Medium auxiliary memory accessible.

---

## 1.5.473 BS\_SCSI\_ERROR\_DISK\_29 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_29 21
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: The disc is not writable.

---

## 1.5.474 BS\_SCSI\_ERROR\_DISK\_30 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_30 22
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Wrong medium.

---

## 1.5.475 BS\_SCSI\_ERROR\_DISK\_31 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_31 23
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: No medium.

## 1.5.476 BS\_SCSI\_ERROR\_DISK\_32 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_32 24
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: The disc is not erasable.

---

## 1.5.477 BS\_SCSI\_ERROR\_DISK\_33 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_33 123
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium error.

---

## 1.5.478 BS\_SCSI\_ERROR\_DISK\_34 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_34 456
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Erase ([see page 153](#)) disc before writing.

---

## 1.5.479 BS\_SCSI\_ERROR\_DISK\_35 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_35 459
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: The disc is not writable with current write method.

## 1.5.480 BS\_SCSI\_ERROR\_DISK\_36 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_36 467
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Cannot write medium - unsupported medium version.

---

## 1.5.481 BS\_SCSI\_ERROR\_DISK\_37 Macro

**C++**

```
#define BS_SCSI_ERROR_DISK_37 477
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Medium is not writeable.

---

## 1.5.482 BS\_SCSI\_ERROR\_DRIVE\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_DRIVE_01 253
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error reading UPC / EAN number.

---

## 1.5.483 BS\_SCSI\_ERROR\_DRIVE\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_DRIVE_02 254
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error reading ISRC number

## 1.5.484 BS\_SCSI\_ERROR\_EXT\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_EXT_01 240
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error detected by third party temporary initiator.

---

## 1.5.485 BS\_SCSI\_ERROR\_EXT\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_EXT_02 241
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Third party device failure.

---

## 1.5.486 BS\_SCSI\_ERROR\_LOG\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_LOG_01 231
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error log overflow.

---

## 1.5.487 BS\_SCSI\_ERROR\_LOG\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_LOG_02 418
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Log exception.

## 1.5.488 BS\_SCSI\_ERROR\_LOG\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_LOG_03 420
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Log counter at maximum.

---

## 1.5.489 BS\_SCSI\_ERROR\_LOG\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_LOG_04 421
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Log list codes exhausted.

---

## 1.5.490 BS\_SCSI\_ERROR\_MECH\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_MECH_01 259
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Mechanical positioning error.

---

## 1.5.491 BS\_SCSI\_ERROR\_MECH\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_MECH_02 260
```

**File**

IsoSDKDefinitions.h ( see page 517)

**Description**

IsoSDK error code: Positioning error detected by read of medium.

## 1.5.492 BS\_SCSI\_ERROR\_MECH\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_MECH_03 366
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Mechanical positioning or changer error.

---

## 1.5.493 BS\_SCSI\_ERROR\_PARAM\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_01 288
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Invalid field in parameter list.

---

## 1.5.494 BS\_SCSI\_ERROR\_PARAM\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_02 289
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Parameter not supported.

---

## 1.5.495 BS\_SCSI\_ERROR\_PARAM\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_03 290
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Parameter value invalid.

## 1.5.496 BS\_SCSI\_ERROR\_PARAM\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_04 291
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Threshold parameters not supported.

---

## 1.5.497 BS\_SCSI\_ERROR\_PARAM\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_05 318
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Parameters changed.

---

## 1.5.498 BS\_SCSI\_ERROR\_PARAM\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_06 319
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Mode parameters changed.

---

## 1.5.499 BS\_SCSI\_ERROR\_PARAM\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_07 320
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Log parameters changed.

## 1.5.500 BS\_SCSI\_ERROR\_PARAM\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_08 351
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Rounded parameter.

---

## 1.5.501 BS\_SCSI\_ERROR\_PARAM\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_09 352
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Saving parameters not supported.

---

## 1.5.502 BS\_SCSI\_ERROR\_PARAM\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_PARAM_10 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: A bad parameter has been passed in.

---

## 1.5.503 BS\_SCSI\_ERROR\_READ\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_READ_01 246
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Unrecovered read error.

## 1.5.504 BS\_SCSI\_ERROR\_READ\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_READ_02 247
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Read retries exhausted.

---

## 1.5.505 BS\_SCSI\_ERROR\_RECOVER\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_01 261
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Recovered data with no error correction applied.

---

## 1.5.506 BS\_SCSI\_ERROR\_RECOVER\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_02 262
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Recovered data with retries.

---

## 1.5.507 BS\_SCSI\_ERROR\_RECOVER\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_03 263
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Recovered data with positive head offset.

## 1.5.508 BS\_SCSI\_ERROR\_RECOVER\_04 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_04 264
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data with negative head offset.

---

## 1.5.509 BS\_SCSI\_ERROR\_RECOVER\_05 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_05 265
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data with retries and/or CIRC applied.

---

## 1.5.510 BS\_SCSI\_ERROR\_RECOVER\_06 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_06 266
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data using previous sector id.

---

## 1.5.511 BS\_SCSI\_ERROR\_RECOVER\_07 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_07 267
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data without ecc - recommend reassignment.

## 1.5.512 BS\_SCSI\_ERROR\_RECOVER\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_08 268
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data without ecc - recommend rewrite.

---

## 1.5.513 BS\_SCSI\_ERROR\_RECOVER\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_09 269
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data without ecc - data rewritten.

---

## 1.5.514 BS\_SCSI\_ERROR\_RECOVER\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_10 270
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data with error correction applied.

---

## 1.5.515 BS\_SCSI\_ERROR\_RECOVER\_11 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_11 271
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data with error corr. & retries applied.

## 1.5.516 BS\_SCSI\_ERROR\_RECOVER\_12 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_12 272
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data - data auto-reallocated.

---

## 1.5.517 BS\_SCSI\_ERROR\_RECOVER\_13 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_13 273
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data with CIRC.

---

## 1.5.518 BS\_SCSI\_ERROR\_RECOVER\_14 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_14 274
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data with I-ec.

---

## 1.5.519 BS\_SCSI\_ERROR\_RECOVER\_15 Macro

### C++

```
#define BS_SCSI_ERROR_RECOVER_15 275
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Recovered data - recommend reassignment.

## 1.5.520 BS\_SCSI\_ERROR\_RECOVER\_16 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_16 276
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data - recommend rewrite.

---

## 1.5.521 BS\_SCSI\_ERROR\_RECOVER\_17 Macro

**C++**

```
#define BS_SCSI_ERROR_RECOVER_17 277
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Recovered data with linking.

---

## 1.5.522 BS\_SCSI\_ERROR\_SEGM\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_SEGM_01 296
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Too many segment descriptors.

---

## 1.5.523 BS\_SCSI\_ERROR\_SEGM\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_SEGM_02 297
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Unsupported segment descriptor type code.

## 1.5.524 BS\_SCSI\_ERROR\_SEGM\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_SEGM_03 298
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Unexpected inexact segment.

---

## 1.5.525 BS\_SCSI\_ERROR\_SEGM\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_SEGM_04 301
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Copy segment granularity violation.

---

## 1.5.526 BS\_SCSI\_ERROR\_SESSION\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_SESSION_01 443
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Session fixation error.

---

## 1.5.527 BS\_SCSI\_ERROR\_SESSION\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_SESSION_02 444
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Session fixation error writing lead-in.

## 1.5.528 BS\_SCSI\_ERROR\_SESSION\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_SESSION_03 445
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Session fixation error writing lead-out.

---

## 1.5.529 BS\_SCSI\_ERROR\_SESSION\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_SESSION_04 446
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Session fixation error - incomplete track in session.

---

## 1.5.530 BS\_SCSI\_ERROR\_TARGET\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_TARGET_01 242
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Copy target device not reachable.

---

## 1.5.531 BS\_SCSI\_ERROR\_TARGET\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_TARGET_02 243
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Incorrect copy target device type.

## 1.5.532 BS\_SCSI\_ERROR\_TARGET\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_TARGET_03 244
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Copy target device data underrun.

---

## 1.5.533 BS\_SCSI\_ERROR\_TARGET\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_TARGET_04 245
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Session fixation error - incomplete track in session.

---

## 1.5.534 BS\_SCSI\_ERROR\_UNIT\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_01 209
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: No seek complete.

---

## 1.5.535 BS\_SCSI\_ERROR\_UNIT\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_02 210
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, cause not reportable.

## 1.5.536 BS\_SCSI\_ERROR\_UNIT\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_03 211
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit is in process of becoming ready.

---

## 1.5.537 BS\_SCSI\_ERROR\_UNIT\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_04 212
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, initializing command required.

---

## 1.5.538 BS\_SCSI\_ERROR\_UNIT\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_05 213
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, manual intervention required.

---

## 1.5.539 BS\_SCSI\_ERROR\_UNIT\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_06 214
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, format in progress.

## 1.5.540 BS\_SCSI\_ERROR\_UNIT\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_07 215
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, operation in progress.

---

## 1.5.541 BS\_SCSI\_ERROR\_UNIT\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_08 216
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, long write in progress.

---

## 1.5.542 BS\_SCSI\_ERROR\_UNIT\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_09 217
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit not ready, self-test in progress.

---

## 1.5.543 BS\_SCSI\_ERROR\_UNIT\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_10 218
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit does not respond to selection.

## 1.5.544 BS\_SCSI\_ERROR\_UNIT\_11 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_11 221
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit communication failure.

---

## 1.5.545 BS\_SCSI\_ERROR\_UNIT\_12 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_12 222
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit communication time-out.

---

## 1.5.546 BS\_SCSI\_ERROR\_UNIT\_13 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_13 223
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit communication parity error.

---

## 1.5.547 BS\_SCSI\_ERROR\_UNIT\_14 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_14 224
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit communication CRC error (Ultra-DMA/32).

## 1.5.548 BS\_SCSI\_ERROR\_UNIT\_16 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_16 287
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Logical unit not supported.

---

## 1.5.549 BS\_SCSI\_ERROR\_UNIT\_17 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_17 368
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Logical unit has not self-configured yet.

---

## 1.5.550 BS\_SCSI\_ERROR\_UNIT\_18 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_18 369
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Logical unit failure.

---

## 1.5.551 BS\_SCSI\_ERROR\_UNIT\_19 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_19 370
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Timeout on logical unit.

## 1.5.552 BS\_SCSI\_ERROR\_UNIT\_20 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_20 371
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit failed self-test.

---

## 1.5.553 BS\_SCSI\_ERROR\_UNIT\_21 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_21 372
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit unable to update self-test log.

---

## 1.5.554 BS\_SCSI\_ERROR\_UNIT\_22 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_22 404
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit failed self-configuration.

---

## 1.5.555 BS\_SCSI\_ERROR\_UNIT\_23 Macro

**C++**

```
#define BS_SCSI_ERROR_UNIT_23 424
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Logical unit failure prediction threshold exceeded.

## 1.5.556 BS\_SCSI\_ERROR\_VOL\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_VOL_01 383
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Volume set created or modified.

---

## 1.5.557 BS\_SCSI\_ERROR\_VOL\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_VOL_02 384
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Volume set deleted.

---

## 1.5.558 BS\_SCSI\_ERROR\_VOL\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_VOL_03 385
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Volume set de-assigned.

---

## 1.5.559 BS\_SCSI\_ERROR\_VOL\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_VOL_04 386
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Volume set reassigned.

## 1.5.560 BS\_SCSI\_ERROR\_WRITE\_01 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_01 235
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: General write error.

---

## 1.5.561 BS\_SCSI\_ERROR\_WRITE\_02 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_02 236
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Write error - Recovery needed.

---

## 1.5.562 BS\_SCSI\_ERROR\_WRITE\_03 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_03 237
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Write error - Recovery failed.

---

## 1.5.563 BS\_SCSI\_ERROR\_WRITE\_04 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_04 238
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Write error - Loss of streaming.

## 1.5.564 BS\_SCSI\_ERROR\_WRITE\_05 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_05 239
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Write error - Padding blocks added.

---

## 1.5.565 BS\_SCSI\_ERROR\_WRITE\_06 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_06 302
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Write protected.

---

## 1.5.566 BS\_SCSI\_ERROR\_WRITE\_07 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_07 303
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Hardware write protected.

---

## 1.5.567 BS\_SCSI\_ERROR\_WRITE\_08 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_08 305
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Associated write protect.

## 1.5.568 BS\_SCSI\_ERROR\_WRITE\_09 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_09 306
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Persistent write protect.

---

## 1.5.569 BS\_SCSI\_ERROR\_WRITE\_10 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_10 307
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Permanent write protect.

---

## 1.5.570 BS\_SCSI\_ERROR\_WRITE\_11 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_11 308
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Conditional write protect.

---

## 1.5.571 BS\_SCSI\_ERROR\_WRITE\_12 Macro

**C++**

```
#define BS_SCSI_ERROR_WRITE_12 142
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A write-error occurred.

## 1.5.572 BS\_SDK\_COMPENC\_BOTH Macro

**C++**

```
#define BS_SDK_COMPENC_BOTH 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium Information: disc or image is compressed and encrypted.

---

## 1.5.573 BS\_SDK\_COMPENC\_COMPRESSED Macro

**C++**

```
#define BS_SDK_COMPENC_COMPRESSED 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium Information: disc or image is compressed.

---

## 1.5.574 BS\_SDK\_COMPENC\_ENCRYPTED Macro

**C++**

```
#define BS_SDK_COMPENC_ENCRYPTED 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Medium Information: disc or image is encrypted.

---

## 1.5.575 BS\_SDK\_CUE\_ERROR\_COMMAND\_01 Macro

**C++**

```
#define BS_SDK_CUE_ERROR_COMMAND_01 160
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid command.

## 1.5.576 BS\_SDK\_CUE\_ERROR\_COMMAND\_06 Macro

**C++**

```
#define BS_SDK_CUE_ERROR_COMMAND_06 13
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Command failed.

---

## 1.5.577 BS\_SDK\_CUE\_ERROR\_FIELD Macro

**C++**

```
#define BS_SDK_CUE_ERROR_FIELD 158
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Invalid field.

---

## 1.5.578 BS\_SDK\_CUE\_ERROR\_FILE Macro

**C++**

```
#define BS_SDK_CUE_ERROR_FILE 159
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: File not found.

---

## 1.5.579 BS\_SDK\_CUE\_ERROR\_SENDING\_CUE Macro

**C++**

```
#define BS_SDK_CUE_ERROR_SENDING_CUE 156
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Send cue sheet failed.

## 1.5.580 BS\_SDK\_CUE\_ERROR\_UEOL Macro

### C++

```
#define BS_SDK_CUE_ERROR_UEOL 157
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Unexpected end of line.

---

## 1.5.581 BS\_SDK\_ERROR\_ABORTED Macro

### C++

```
#define BS_SDK_ERROR_ABORTED 138
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Operation aborted by user.

---

## 1.5.582 BS\_SDK\_ERROR\_ABORTEDCOMMAND Macro

### C++

```
#define BS_SDK_ERROR_ABORTEDCOMMAND 130
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Interface reported aborted command was thrown.

---

## 1.5.583 BS\_SDK\_ERROR\_BAD\_CAPABILITY\_NAME Macro

### C++

```
#define BS_SDK_ERROR_BAD_CAPABILITY_NAME 491
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

Capabilities error: A not existing capability name was submitted.

## 1.5.584 BS\_SDK\_ERROR\_BAD\_REQUEST Macro

**C++**

```
#define BS_SDK_ERROR_BAD_REQUEST 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Bad request.

---

## 1.5.585 BS\_SDK\_ERROR\_BIN\_FILE\_NOT\_FOUND Macro

**C++**

```
#define BS_SDK_ERROR_BIN_FILE_NOT_FOUND 18
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The bin file could not be found.

---

## 1.5.586 BS\_SDK\_ERROR\_BUFFERALIGNMENT Macro

**C++**

```
#define BS_SDK_ERROR_BUFFERALIGNMENT 105
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Buffer alignment error.

---

## 1.5.587 BS\_SDK\_ERROR\_BUFFERTOOBIG Macro

**C++**

```
#define BS_SDK_ERROR_BUFFERTOOBIG 107
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The buffer is too big.

---

## 1.5.588 BS\_SDK\_ERROR\_BURN\_IN\_PROGRESS Macro

**C++**

```
#define BS_SDK_ERROR_BURN_IN_PROGRESS 27
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Not allowed while burning.

---

## 1.5.589 BS\_SDK\_ERROR\_CDTEXT\_NOT\_FOUND Macro

**C++**

```
#define BS_SDK_ERROR_CDTEXT_NOT_FOUND 176
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: CDText is not available on the audio disc.

---

## 1.5.590 BS\_SDK\_ERROR\_CHECKCONDITION Macro

**C++**

```
#define BS_SDK_ERROR_CHECKCONDITION 117
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Check condition.

---

## 1.5.591 BS\_SDK\_ERROR\_COMPENC\_GENERALERROR Macro

**C++**

```
#define BS_SDK_ERROR_COMPENC_GENERALERROR 497
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: General error while read compress/encrypt identifier.

---

## 1.5.592 BS\_SDK\_ERROR\_COMPENC\_READ10\_IDENT Macro

**C++**

```
#define BS_SDK_ERROR_COMPENC_READ10_IDENT 498
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: No IsoSDK identifier found on disc or image.

---

## 1.5.593 BS\_SDK\_ERROR\_COMPRESSION\_CONFLICT Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_CONFLICT 166
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: A compression conflict occurred.

---

## 1.5.594 BS\_SDK\_ERROR\_COMPRESSION\_NOINDXTABLE Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_NOINDXTABLE 496
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: No Indexes table found.

---

## 1.5.595 BS\_SDK\_ERROR\_COMPRESSION\_NOSIGNATURE Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_NOSIGNATURE 493
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: No IsoSDK compression signature found.

---

## 1.5.596 BS\_SDK\_ERROR\_COMPRESSION\_READ10 Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_READ10 492
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error while operate Read10 from image or disc. Maybe not data.

---

## 1.5.597 BS\_SDK\_ERROR\_COMPRESSION\_READMEMF Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_READMEMF 495
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error while read data from MemFile. Maybe empty.

---

## 1.5.598 BS\_SDK\_ERROR\_COMPRESSION\_SEEKMEMFILE Macro

**C++**

```
#define BS_SDK_ERROR_COMPRESSION_SEEKMEMFILE 494
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error while seek inside MemFile. Maybe no data.

## 1.5.599 BS\_SDK\_ERROR\_COPYABORTED Macro

### C++

```
#define BS_SDK_ERROR_COPYABORTED 129
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Copy aborted.

---

## 1.5.600

## BS\_SDK\_ERROR\_CORRUPT\_OR\_INVALID\_CUE\_FILE Macro

### C++

```
#define BS_SDK_ERROR_CORRUPT_OR_INVALID_CUE_FILE 17
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The cue file is corrupt or completely invalid.

---

## 1.5.601 BS\_SDK\_ERROR\_CREATEFILE Macro

### C++

```
#define BS_SDK_ERROR_CREATEFILE 140
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: IsoSDK was not able to create the file.

---

## 1.5.602 BS\_SDK\_ERROR\_DATAOVERRUN Macro

### C++

```
#define BS_SDK_ERROR_DATAOVERRUN 115
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: A data-overrun occurred.

---

## 1.5.603 BS\_SDK\_ERROR\_DATAPROTECT Macro

**C++**

```
#define BS_SDK_ERROR_DATAPROTECT 127
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Data protected.

---

## 1.5.604 BS\_SDK\_ERROR\_DATARECOVERERROR Macro

**C++**

```
#define BS_SDK_ERROR_DATARECOVERERROR 121
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: The IsoSDK was not able to recover the data.

---

## 1.5.605 BS\_SDK\_ERROR\_DEVICE\_LOCKED Macro

**C++**

```
#define BS_SDK_ERROR_DEVICE_LOCKED 183
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: The device is locked. Maybe by another software / process.

---

## 1.5.606 BS\_SDK\_ERROR\_EMPTY\_PASSWORD Macro

**C++**

```
#define BS_SDK_ERROR_EMPTY_PASSWORD 29
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Empty passwords are not allowed.

## 1.5.607 BS\_SDK\_ERROR\_ENCLIB\_NOT\_FOUND Macro

### C++

```
#define BS_SDK_ERROR_ENCLIB_NOT_FOUND 175
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: bass library was not found in directory.

---

## 1.5.608 BS\_SDK\_ERROR\_ENCRYPTION\_CONFLICT Macro

### C++

```
#define BS_SDK_ERROR_ENCRYPTION_CONFLICT 167
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Encryption conflict.

---

## 1.5.609 BS\_SDK\_ERROR\_ERASECHECK Macro

### C++

```
#define BS_SDK_ERROR_ERASECHECK 128
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Erase ([see page 153](#)) check.

---

## 1.5.610 BS\_SDK\_ERROR\_ERRINVALIDFILENAME Macro

### C++

```
#define BS_SDK_ERROR_ERRINVALIDFILENAME 148
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Error importing session, file name length exceeds 120 chars.

## 1.5.611 BS\_SDK\_ERROR\_FILE\_EXISTS Macro

**C++**

```
#define BS_SDK_ERROR_FILE_EXISTS 12
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The file already exists.

---

## 1.5.612 BS\_SDK\_ERROR\_FILE\_OPEN Macro

**C++**

```
#define BS_SDK_ERROR_FILE_OPEN 16
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: File could not be opened.

---

## 1.5.613 BS\_SDK\_ERROR\_FILEINUSE Macro

**C++**

```
#define BS_SDK_ERROR_FILEINUSE 139
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: File could not be opened.

---

## 1.5.614 BS\_SDK\_ERROR\_FILEMARK Macro

**C++**

```
#define BS_SDK_ERROR_FILEMARK 134
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Filemark error.

## 1.5.615 BS\_SDK\_ERROR\_GENERAL Macro

**C++**

```
#define BS_SDK_ERROR_GENERAL 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: A general error occurred.

---

## 1.5.616 BS\_SDK\_ERROR\_HARDWAREERROR Macro

**C++**

```
#define BS_SDK_ERROR_HARDWAREERROR 124
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Hardware error.

---

## 1.5.617 BS\_SDK\_ERROR\_ILLEGALLENGTH Macro

**C++**

```
#define BS_SDK_ERROR_ILLEGALLENGTH 136
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Illegal length of file name or path.

---

## 1.5.618 BS\_SDK\_ERROR\_IMPORTSESSION Macro

**C++**

```
#define BS_SDK_ERROR_IMPORTSESSION 149
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Error importing session.

## 1.5.619 BS\_SDK\_ERROR\_INCOMPATIBLE\_FS\_TYPE Macro

### C++

```
#define BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE 165
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Selected project type is incompatible with last recorded session.

---

## 1.5.620 BS\_SDK\_ERROR\_INCORRECTLENGTH Macro

### C++

```
#define BS_SDK_ERROR_INCORRECTLENGTH 137
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Incorrect length of the given file name or path.

---

## 1.5.621 BS\_SDK\_ERROR\_INVALID\_DEST\_PATH Macro

### C++

```
#define BS_SDK_ERROR_INVALID_DEST_PATH 9
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: An invalid destination path has been passed in.

---

## 1.5.622 BS\_SDK\_ERROR\_INVALID\_DIR\_INDEX Macro

### C++

```
#define BS_SDK_ERROR_INVALID_DIR_INDEX 168
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Invalid directory index.

## 1.5.623 BS\_SDK\_ERROR\_INVALID\_FILE\_FORMAT Macro

### C++

```
#define BS_SDK_ERROR_INVALID_FILE_FORMAT 15
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The file has an invalid format.

---

## 1.5.624 BS\_SDK\_ERROR\_INVALID\_FILE\_NAME Macro

### C++

```
#define BS_SDK_ERROR_INVALID_FILE_NAME 10
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: An invalid file name has been passed in.

---

## 1.5.625 BS\_SDK\_ERROR\_INVALID\_HANDLE Macro

### C++

```
#define BS_SDK_ERROR_INVALID_HANDLE 490
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The submitted handle is not a capability handle or was not initialized with GetDeviceCapabilitiesHandle ([see page 162](#)()).

---

## 1.5.626 BS\_SDK\_ERROR\_INVALID\_INDEX Macro

### C++

```
#define BS_SDK_ERROR_INVALID_INDEX 173
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The SubIndex of the Audiotrack is wrong.

## 1.5.627 BS\_SDK\_ERROR\_INVALID\_ISRC Macro

### C++

```
#define BS_SDK_ERROR_INVALID_ISRC 170
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The ISRC code inside the track information is invalid.

---

## 1.5.628 BS\_SDK\_ERROR\_INVALID\_MCN Macro

### C++

```
#define BS_SDK_ERROR_INVALID_MCN 169
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The MCN code inside the track information is invalid.

---

## 1.5.629 BS\_SDK\_ERROR\_INVALID\_PATH Macro

### C++

```
#define BS_SDK_ERROR_INVALID_PATH 7
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: An invalid path has been passed in.

---

## 1.5.630 BS\_SDK\_ERROR\_INVALID\_SESSION\_NUMBER Macro

### C++

```
#define BS_SDK_ERROR_INVALID_SESSION_NUMBER 28
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Invalid session number has been passed in.

## 1.5.631 BS\_SDK\_ERROR\_INVALID\_SRC\_PATH Macro

### C++

```
#define BS_SDK_ERROR_INVALID_SRC_PATH 8
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: An invalid source path has been passed in.

---

## 1.5.632 BS\_SDK\_ERROR\_INVALID\_UDF\_VERSION Macro

### C++

```
#define BS_SDK_ERROR_INVALID_UDF_VERSION 163
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Invalid UDF version number.

---

## 1.5.633 BS\_SDK\_ERROR\_INVALIDSRB Macro

### C++

```
#define BS_SDK_ERROR_INVALIDSRB 104
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Invalid SRB.

---

## 1.5.634 BS\_SDK\_ERROR\_ISOIMAGENOTFOUND Macro

### C++

```
#define BS_SDK_ERROR_ISOIMAGENOTFOUND 150
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Error open ISO image file. ISO image not found.

## 1.5.635 BS\_SDK\_ERROR\_MAXDIRS Macro

**C++**

```
#define BS_SDK_ERROR_MAXDIRS 146
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Not more than %d directories are allowed.

---

## 1.5.636 BS\_SDK\_ERROR\_MAXFILES Macro

**C++**

```
#define BS_SDK_ERROR_MAXFILES 147
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Not more than %d files are allowed.

---

## 1.5.637 BS\_SDK\_ERROR\_MESSAGEJECT Macro

**C++**

```
#define BS_SDK_ERROR_MESSAGEJECT 110
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Message has been rejected.

---

## 1.5.638 BS\_SDK\_ERROR\_MISCOMPARE Macro

**C++**

```
#define BS_SDK_ERROR_MISCOMPARE 132
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Miscompare. Compared names are different.

## 1.5.639 BS\_SDK\_ERROR\_MORE\_SPACE\_NEEDED Macro

### C++

```
#define BS_SDK_ERROR_MORE_SPACE_NEEDED 26
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The supplied buffer is too small.

---

## 1.5.640 BS\_SDK\_ERROR\_MP3LIB\_NOT\_FOUND Macro

### C++

```
#define BS_SDK_ERROR_MP3LIB_NOT_FOUND 164
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: mp3lib (FoxPlayer) not found.

---

## 1.5.641 BS\_SDK\_ERROR\_NETTAGS\_CONNECT Macro

### C++

```
#define BS_SDK_ERROR_NETTAGS_CONNECT 178
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Error while connect to the CDDB server.

---

## 1.5.642 BS\_SDK\_ERROR\_NETTAGS\_DISK Macro

### C++

```
#define BS_SDK_ERROR_NETTAGS_DISK 177
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Compact disc error during forming request to internet CDDB tags.

## 1.5.643 BS\_SDK\_ERROR\_NETTAGS\_INTERNAL Macro

**C++**

```
#define BS_SDK_ERROR_NETTAGS_INTERNAL 181
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A internal error occurred. Please contact our support.

---

## 1.5.644 BS\_SDK\_ERROR\_NETTAGS\_NOMATCH Macro

**C++**

```
#define BS_SDK_ERROR_NETTAGS_NOMATCH 180
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: No match while searching CDDB database.

---

## 1.5.645 BS\_SDK\_ERROR\_NETTAGS\_SERVER Macro

**C++**

```
#define BS_SDK_ERROR_NETTAGS_SERVER 179
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A server error occur. With FreeDB try another mirror.

---

## 1.5.646 BS\_SDK\_ERROR\_NEXTADDRESS Macro

**C++**

```
#define BS_SDK_ERROR_NEXTADDRESS 144
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error getting next writable address.

## 1.5.647 BS\_SDK\_ERROR\_NO Macro

### C++

```
#define BS_SDK_ERROR_NO 0
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: No error occurred.

---

## 1.5.648 BS\_SDK\_ERROR\_NOT\_ALLOWED Macro

### C++

```
#define BS_SDK_ERROR_NOT_ALLOWED 1
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Operation / command not allowed.

---

## 1.5.649

## BS\_SDK\_ERROR\_NOT\_ALLOWED\_FOR\_THIS\_BURNER Macro

### C++

```
#define BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER 20
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Not allowed for the current burner.

---

## 1.5.650

## BS\_SDK\_ERROR\_NOT\_ALLOWED\_FOR\_THIS\_PROJECTTYPE Macro new

### C++

```
#define BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE 808
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: If you set a forbidden option or function according to the used project.

---

## 1.5.651

# BS\_SDK\_ERROR\_NOT\_ALLOWED\_FOR\_THIS\_UDF\_VERSION Macro

**C++**

```
#define BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION 162
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Invalid option for this UDF version.

---

## 1.5.652 BS\_SDK\_ERROR\_NOT\_IMPLEMENTED Macro

**C++**

```
#define BS_SDK_ERROR_NOT_IMPLEMENTED 19
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The feature ( command is currently not implemented.

---

## 1.5.653 BS\_SDK\_ERROR\_NOTREADY Macro

**C++**

```
#define BS_SDK_ERROR_NOTREADY 122
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: Device not ready.

## 1.5.654 BS\_SDK\_ERROR\_NOTSUPPORTED Macro

### C++

```
#define BS_SDK_ERROR_NOTSUPPORTED 143
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Operation / command not supported.

---

## 1.5.655 BS\_SDK\_ERROR\_PARITYERR Macro

### C++

```
#define BS_SDK_ERROR_PARITYERR 112
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Partition version error.

---

## 1.5.656 BS\_SDK\_ERROR\_PATH\_EXISTS Macro

### C++

```
#define BS_SDK_ERROR_PATH_EXISTS 11
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: The path already exists.

---

## 1.5.657 BS\_SDK\_ERROR\_PLUGIN Macro

### C++

```
#define BS_SDK_ERROR_PLUGIN 184
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Error while loading external plugin.

## 1.5.658 BS\_SDK\_ERROR\_RESERVED Macro

**C++**

```
#define BS_SDK_ERROR_RESERVED 133
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Reserved.

---

## 1.5.659 BS\_SDK\_ERROR\_SELECTIONTIMEOUT Macro

**C++**

```
#define BS_SDK_ERROR_SELECTIONTIMEOUT 114
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Selection timed out.

---

## 1.5.660 BS\_SDK\_ERROR\_SRBTIMEOUT Macro

**C++**

```
#define BS_SDK_ERROR_SRBTIMEOUT 109
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A SRB timeout occurred.

---

## 1.5.661 BS\_SDK\_ERROR\_TIMEOUT Macro

**C++**

```
#define BS_SDK_ERROR_TIMEOUT 108
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A timeout occurred.

## 1.5.662 BS\_SDK\_ERROR\_TOO MUCH DATA Macro

### C++

```
#define BS_SDK_ERROR_TOO MUCH DATA 145
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Too much data for this mode. The size of the collection will exceed the disc size.

---

## 1.5.663 BS\_SDK\_ERROR\_TOO MUCH INDEXES Macro

### C++

```
#define BS_SDK_ERROR_TOO MUCH INDEXES 174
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: To much SubIndex values found. Max is 99.

---

## 1.5.664 BS\_SDK\_ERROR\_UNEXPECTEDBUSFREE Macro

### C++

```
#define BS_SDK_ERROR_UNEXPECTEDBUSFREE 116
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Unexpected bus free.

---

## 1.5.665 BS\_SDK\_ERROR\_UNITATTENTION Macro

### C++

```
#define BS_SDK_ERROR_UNITATTENTION 126
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK error code: Unit attention.

## 1.5.666 BS\_SDK\_ERROR\_UNKNOWN Macro

**C++**

```
#define BS_SDK_ERROR_UNKNOWN 101
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: An unknown error occurred.

---

## 1.5.667 BS\_SDK\_ERROR\_UNKNOWN\_TEXTID Macro

**C++**

```
#define BS_SDK_ERROR_UNKNOWN_TEXTID 25
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: The text ID that has been passed in is unknown.

---

## 1.5.668 BS\_SDK\_ERROR\_UNSUPPORTED\_MEDIUM Macro

**C++**

```
#define BS_SDK_ERROR_UNSUPPORTED_MEDIUM 182
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: A medium was inserted that is not supported by IsoSDK.

---

## 1.5.669 BS\_SDK\_ERROR\_VOLUME\_OVERFLOW Macro

**C++**

```
#define BS_SDK_ERROR_VOLUME_OVERFLOW 131
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Volume overflow.

## 1.5.670 BS\_SDK\_INT\_ERROR\_1 Macro

**C++**

```
#define BS_SDK_INT_ERROR_1 151
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Internal error 1.

---

## 1.5.671 BS\_SDK\_INT\_ERROR\_2 Macro

**C++**

```
#define BS_SDK_INT_ERROR_2 152
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Internal error 2.

---

## 1.5.672 BS\_SDK\_INT\_ERROR\_3 Macro

**C++**

```
#define BS_SDK_INT_ERROR_3 153
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Internal error 3.

---

## 1.5.673 BS\_SDK\_INT\_ERROR\_4 Macro

**C++**

```
#define BS_SDK_INT_ERROR_4 154
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Internal error 4.

## 1.5.674 BS\_SDK\_INT\_ERROR\_5 Macro

**C++**

```
#define BS_SDK_INT_ERROR_5 155
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Internal error 5.

---

## 1.5.675 BS\_SDK\_INT\_ERROR\_FORMAT Macro

**C++**

```
#define BS_SDK_INT_ERROR_FORMAT 161
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK error code: Error formatting re-writeable medium.

---

## 1.5.676 BS\_SDK\_KEY\_INVALID Macro

**C++**

```
#define BS_SDK_KEY_INVALID 807
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK message: An invalid license key was submitted.

---

## 1.5.677 BS\_SDK\_KEY\_VALID Macro

**C++**

```
#define BS_SDK_KEY_VALID 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK message: The license key is valid.

## 1.5.678 BS\_SDK\_MESSAGE\_01 Macro

### C++

```
#define BS_SDK_MESSAGE_01 700
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK Message: Text unknown

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.679 BS\_SDK\_MESSAGE\_02 Macro

### C++

```
#define BS_SDK_MESSAGE_02 701
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Waiting for user interaction

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.680 BS\_SDK\_MESSAGE\_03 Macro

### C++

```
#define BS_SDK_MESSAGE_03 702
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Cannot close dialog

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.681 BS\_SDK\_MESSAGE\_04 Macro

### C++

```
#define BS_SDK_MESSAGE_04 703
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Please stop the process first.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.682 BS\_SDK\_MESSAGE\_05 Macro

### C++

```
#define BS_SDK_MESSAGE_05 704
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Erasing CD/DVD ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.683 BS\_SDK\_MESSAGE\_06 Macro

### C++

```
#define BS_SDK_MESSAGE_06 705
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Preparing Data ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.684 BS\_SDK\_MESSAGE\_07 Macro

### C++

```
#define BS_SDK_MESSAGE_07 706
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Finalizing CD/DVD ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.685 BS\_SDK\_MESSAGE\_08 Macro

### C++

```
#define BS_SDK_MESSAGE_08 707
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Burning CD/DVD ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.686 BS\_SDK\_MESSAGE\_10 Macro

### C++

```
#define BS_SDK_MESSAGE_10 708
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Aborting Process ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.687 BS\_SDK\_MESSAGE\_11 Macro

### C++

```
#define BS_SDK_MESSAGE_11 709
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Process successfully completed.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.688 BS\_SDK\_MESSAGE\_12 Macro

### C++

```
#define BS_SDK_MESSAGE_12 710
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: max.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.689 BS\_SDK\_MESSAGE\_13 Macro

### C++

```
#define BS_SDK_MESSAGE_13 711
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Please insert the next CD/DVD.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.690 BS\_SDK\_MESSAGE\_14 Macro

### C++

```
#define BS_SDK_MESSAGE_14 712
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Starting the verify process ...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.691 BS\_SDK\_MESSAGE\_15 Macro

### C++

```
#define BS_SDK_MESSAGE_15 713
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Process completed with %d error(s).

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.692 BS\_SDK\_MESSAGE\_16 Macro

### C++

```
#define BS_SDK_MESSAGE_16 714
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Verifying file.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.693 BS\_SDK\_MESSAGE\_COMPENC\_BOTH Macro new

### C++

```
#define BS_SDK_MESSAGE_COMPENC_BOTH 806
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The IsoSDK will throw this message if the signature for compression and encryption was found.

---

## 1.5.694 BS\_SDK\_MESSAGE\_COMPENC\_COMPRESSED Macro new

### C++

```
#define BS_SDK_MESSAGE_COMPENC_COMPRESSED 805
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The IsoSDK will throw this message if the signature for compression was found.

---

## 1.5.695 BS\_SDK\_MESSAGE\_COMPENC\_ENCRYPTED Macro new

### C++

```
#define BS_SDK_MESSAGE_COMPENC_ENCRYPTED 804
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

The IsoSDK will throw this message if the signature for encryption was found.

---

## 1.5.696 BS\_SDK\_MESSAGE\_ERASESTART Macro

### C++

```
#define BS_SDK_MESSAGE_ERASESTART 602
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Starting the erase process.

**Notes**

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.697 BS\_SDK\_MESSAGE\_EXTR\_FILE Macro

**C++**

```
#define BS_SDK_MESSAGE_EXTR_FILE 603
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK message: Extracting file.

**Notes**

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.698 BS\_SDK\_MESSAGE\_FORMAT Macro

**C++**

```
#define BS_SDK_MESSAGE_FORMAT 606
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK message: Formatting.

**Notes**

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.699 BS\_SDK\_MESSAGE\_FORMAT\_DONE Macro

**C++**

```
#define BS_SDK_MESSAGE_FORMAT_DONE 607
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

IsoSDK message: Formatting is done. Thrown after BS\_SDK\_MESSAGE\_FORMAT ([see page 483](#)).

**Notes**

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.700 BS\_SDK\_MESSAGE\_IMAGECREATESTART Macro

### C++

```
#define BS_SDK_MESSAGE_IMAGECREATESTART 608
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Image creation started.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.701 BS\_SDK\_MESSAGE\_IMPORT Macro

### C++

```
#define BS_SDK_MESSAGE_IMPORT 605
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Importing session.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.702

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_CANNOT\_SELECT\_ENTRY

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY 724
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Database entry can't be selected.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.703

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_ERROR\_CREATING\_MATCHING\_PARAM Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_MATCHING_PARAM 718
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message:

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.704

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_ERROR\_CREATING\_PROVIDER Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_PROVIDER 716
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Error creating tag provider: %s

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.705

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_ERROR\_GETTING\_MATCHING\_ENTRIES

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES 720
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Error getting matching entries: %s

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.706

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_ERROR\_GETTING\_SELECTED\_ENTRY

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY 726
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Error getting selected entry: %s

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.707

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_FETCHING\_SELECTED\_ENTRY Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY 725
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Fetching selected entry, please wait...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.708

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_GETTING\_MATCHING\_ENTRIES Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES 719
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Getting matching entries, please wait...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.709

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_GETTING\_MATCHING\_PARAMS

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS 717
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Getting matching parameters, please wait...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.710

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_INIT\_COMPLETE

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE 715
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK message: Initialization complete. Processing...

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.711

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_MULTIPLE\_MATCHES

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES 721
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Multiple matches.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.712

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_SELECT\_ENTRY

## Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY 723
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Please, select an entry from the list below.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

## 1.5.713

# BS\_SDK\_MESSAGE\_INTERNETDB\_STATUS\_SINGLE\_MATCH Macro

### C++

```
#define BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH 722
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Single match.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.714 BS\_SDK\_MESSAGE\_SIMULATE Macro

### C++

```
#define BS_SDK_MESSAGE_SIMULATE 604
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Test write.

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.715 BS\_SDK\_MESSAGE\_WAIT Macro

### C++

```
#define BS_SDK_MESSAGE_WAIT 600
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

IsoSDK message: Writing lead-out (may take 15-20 minutes).

### Notes

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.716 BS\_SDK\_MESSAGE\_WRITESTART Macro

**C++**

```
#define BS_SDK_MESSAGE_WRITESTART 601
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK message: Starting write process.

**Notes**

Messages are predefined strings inside the IsoSDK. To change the language of this messages you can use this message strings inside a own language file. The messages will be shown according to this message strings.

---

## 1.5.717 BS\_SDK\_VERIFY\_ERROR\_CDFILEUNREADABLE Macro

**C++**

```
#define BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE 801
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The burned file is not readable.

---

## 1.5.718 BS\_SDK\_VERIFY\_ERROR\_FILESDIFFERENT Macro

**C++**

```
#define BS_SDK_VERIFY_ERROR_FILESDIFFERENT 802
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The burned file differs from the source file.

---

## 1.5.719 BS\_SDK\_VERIFY\_ERROR\_HDDFILEUNREADABLE Macro

**C++**

```
#define BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE 800
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK error code: The source file is not readable.

---

## 1.5.720 BS\_SEPARATE\_CHANNEL\_MUTE Macro

**C++**

```
#define BS_SEPARATE_CHANNEL_MUTE 0xFFFF00000014LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support independently muting each audio channel via the CD Audio Control Page.

---

## 1.5.721 BS\_SEPARATE\_VOLUME\_LEVELS Macro

**C++**

```
#define BS_SEPARATE_VOLUME_LEVELS 0xFFFF00000013LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: The Logical Unit shall support separately controllable audio levels for each supported channel via the CD Audio Control Page.

---

## 1.5.722 BS\_SMART Macro

**C++**

```
#define BS_SMART 0x004000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to perform Self-Monitoring Analysis and Reporting Technology.

---

## 1.5.723 BS\_STREAMING Macro

**C++**

```
#define BS_STREAMING 0x040000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to perform reading and writing within specified performance ranges.

---

## 1.5.724 BS\_STRING\_MANIPULATION\_DETECTED Macro new

**C++**

```
#define BS_STRING_MANIPULATION_DETECTED 803
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

The IsoSDK will check the string length inside the function body of dedicated classes. If a wrong string length was found the IsoSDK will assume a manipulation and will stop to avoid attacks.

---

## 1.5.725 BS\_TCH\_CDTEXT Macro

**C++**

```
#define BS_TCH_CDTEXT 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio grabber tag information: Receive tags only from CD-Text.

---

## 1.5.726 BS\_TCH\_CDTEXT\_INTERNETDB Macro

**C++**

```
#define BS_TCH_CDTEXT_INTERNETDB 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio grabber tag information: Receive tags first from CD-Text then from FreeDB / CDDB.

---

## 1.5.727 BS\_TCH\_INTERNETDB Macro

**C++**

```
#define BS_TCH_INTERNETDB 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio grabber tag information: Receive tags only from FreeDB / CDDB

---

## 1.5.728 BS\_TCH\_INTERNETDB\_CDTEXT Macro

**C++**

```
#define BS_TCH_INTERNETDB_CDTEXT 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio grabber tag information: Receive tags First from FreeDB/CDDB then from CD-Text.

---

## 1.5.729 BS\_TCH\_NONE Macro

**C++**

```
#define BS_TCH_NONE 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio grabber tag information: Receive no tags.

---

## 1.5.730 BS\_TCI\_ARTIST Macro

**C++**

```
#define BS_TCI_ARTIST 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Artist information.

---

## 1.5.731 BS\_TCI\_CATEGORY Macro

**C++**

```
#define BS_TCI_CATEGORY 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Category information.

---

## 1.5.732 BS\_TCI\_DISK\_LENGTH Macro

**C++**

```
#define BS_TCI_DISK_LENGTH 7
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio disc CDDB tag information: disc length information

---

## 1.5.733 BS\_TCI\_DISKID Macro

**C++**

```
#define BS_TCI_DISKID 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio disc CDDB tag information: Get the Disc id.

---

## 1.5.734 BS\_TCI\_EXTENDED\_INFO Macro

**C++**

```
#define BS_TCI_EXTENDED_INFO 6
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio disc CDDB tag information: Extended information.

---

## 1.5.735 BS\_TCI\_FRAME\_OFFSET Macro

**C++**

```
#define BS_TCI_FRAME_OFFSET 8
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Audio disc CDDB tag information: Frame offset information.

---

## 1.5.736 BS\_TCI\_GENRE Macro

**C++**

```
#define BS_TCI_GENRE 5
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Genre information.

---

## 1.5.737 BS\_TCI\_REVISION Macro

**C++**

```
#define BS_TCI_REVISION 9
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Revision information.

---

## 1.5.738 BS\_TCI\_SUBMITTED\_VIA Macro

**C++**

```
#define BS_TCI_SUBMITTED_VIA 4
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Receive information about the identifier that was used to submit data.

---

## 1.5.739 BS\_TCI\_TITLE Macro

**C++**

```
#define BS_TCI_TITLE 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Receive track or disc title.

## 1.5.740 BS\_TCI\_YEAR Macro

**C++**

```
#define BS_TCI_YEAR 10
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Audio disc CDDB tag information: Receive production year.

---

## 1.5.741 BS\_TF\_AUDIO Macro

**C++**

```
#define BS_TF_AUDIO 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Track type: Audio track.

---

## 1.5.742 BS\_TF\_DATA\_MODE1 Macro

**C++**

```
#define BS_TF_DATA_MODE1 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Track type: Mode 1 track (always for DVD/BD).

---

## 1.5.743 BS\_TF\_DATA\_MODE2 Macro

**C++**

```
#define BS_TF_DATA_MODE2 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Track type: Mode 2 track.

## 1.5.744 BS\_TM\_DMA Macro

**C++**

```
#define BS_TM_DMA 3
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK transfer mode: Direct Memory Access

---

## 1.5.745 BS\_TM\_NOT\_APPLICABLE Macro

**C++**

```
#define BS_TM_NOT_APPLICABLE 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK transfer mode: Not possible to evaluate or set.

---

## 1.5.746 BS\_TM\_PIO Macro

**C++**

```
#define BS_TM_PIO 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK transfer mode: Programmed Input/Output

---

## 1.5.747 BS\_TM\_UNKNOWN Macro

**C++**

```
#define BS_TM_UNKNOWN 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

IsoSDK transfer mode: Unknown

## 1.5.748 BS\_TRUE Macro

### C++

```
#define BS_TRUE 1
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

IsoSDK BOOL value: TRUE

---

## 1.5.749 BS\_UDF\_PARTITION\_PHYSICAL Macro

### C++

```
#define BS_UDF_PARTITION_PHYSICAL 0
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

UDF partition type: Physical or Type 1

This is the simplest partition. A type 1 partition has a start address S and size N. A logical block number A in the partition can be converted to the media physical address (in UDF's term, the logical sector address) S+A. In certain optical media, the start and size of the partition must be aligned to the packet size (such as 32KB). These special requirements are defined in the appendixes of the UDF standard. Free space of the partition is managed by the Unallocated Space Bitmap Descriptor. It contains one bit for each block of the partition. If the bit is set (1), the corresponding block is free. If it is clear (0), the corresponding block is allocated. This is contrary to what FFS/UFS uses the bitmap, because the bitmap in UDF is called Unallocated Space Bitmap.

---

## 1.5.750 BS\_UDF\_PARTITION\_SPARABLE Macro

### C++

```
#define BS_UDF_PARTITION_SPARABLE 2
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

UDF partition type: Sparable.

Sparable partitions are used on overwrite media that will fail after a certain number of overwrites (several thousands), such as CD-RW. In a file system, the places that are overwritten frequently are often important metadata area, e.g., bitmaps. Sparable partition allows the failed area to be remapped to other good part on the media so the failed area appears good to the upper level.

A sparable partition is similar to a type 1 partition in the sense that it has a start address and size. Moreover, it defines 2 to 4 sparing tables which points to reserved spare area on the media. Each sparing table has identical information. The unit of overwrite on such media is packet. For example, the packet size for CD-RW is 32 2K-sectors. One sector in packet failing means the whole packet fails. When this happens, the content of this packet is written to a spare area, and its new address

is written to the sparing table. When translating a logical address in the sparable partition to the physical address, the sparing table is always consulted. If the logical address is not found in the sparing table, the address translation is the same as a type 1 partition. Otherwise, its new address in the sparing area recorded in the sparing table is returned. Thus, the sparing table acts as an exception table in the address translation. This mechanism guarantees that the logical address does not change when its original packet fails.

---

## 1.5.751 BS\_UDF\_PARTITION\_VIRTUAL Macro

**C++**

```
#define BS_UDF_PARTITION_VIRTUAL 1
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF partition type: Virtual.

Virtual partition is used on write-once media. Only three types of metadata are stored in the virtual partition: File Set Descriptor, File Entry (including Extended File Entry), and Allocation Extent Descriptor. If the file data is embedded in the file entry, these file data are also stored in the virtual partition. Virtual partition makes the write-once media appear as an overwrite media. Virtual partition layers on top of the type 1 partition. A Virtual Allocation Table (VAT) is used to map logical addresses of the virtual partition to logical addresses in the underlying type 1 partition.

---

## 1.5.752 BS\_UDF\_VERSION\_102 Macro

**C++**

```
#define BS_UDF_VERSION_102 0x102
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 1.02 - VideoDVD

---

## 1.5.753 BS\_UDF\_VERSION\_150 Macro

**C++**

```
#define BS_UDF_VERSION_150 0x150
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 1.50

## 1.5.754 BS\_UDF\_VERSION\_200 Macro

**C++**

```
#define BS_UDF_VERSION_200 0x200
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 2.00

---

## 1.5.755 BS\_UDF\_VERSION\_201 Macro

**C++**

```
#define BS_UDF_VERSION_201 0x201
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 2.01

---

## 1.5.756 BS\_UDF\_VERSION\_250 Macro

**C++**

```
#define BS_UDF_VERSION_250 0x250
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 2.50

**Notes**

Important: UDF 2.5 and UDF 2.6 is not readable by Windows XP because Windows XP do not contain the driver for this UDF versions.

---

## 1.5.757 BS\_UDF\_VERSION\_260 Macro

**C++**

```
#define BS_UDF_VERSION_260 0x260
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

UDF version: 2.60

**Notes**

Important: UDF 2.5 and UDF 2.6 is not readable by Windows XP because Windows XP do not contain the driver for this UDF versions.

---

## 1.5.758 BS\_UDFOPT\_ALL Macro

**C++**

```
#define BS_UDFOPT_ALL 0xFFFFFFFF
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Set all options.

---

## 1.5.759 BS\_UDFOPT\_AVCHD\_DISC Macro

**C++**

```
#define BS_UDFOPT_AVCHD_DISC 0x00000008 // SUDFOptionsEx.AvchdDisc is valid
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Set AVCHD flag.

---

## 1.5.760 BS\_UDFOPT\_FILE\_STREAMS Macro

**C++**

```
#define BS_UDFOPT_FILE_STREAMS 0x00000004 // SUDFOptionsEx.WriteFileStreams is valid
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Set write file streams.

---

## 1.5.761 BS\_UDFOPT\_IMPLEMENTATION\_ID Macro

**C++**

```
#define BS_UDFOPT_IMPLEMENTATION_ID 0x00000010 // SUDFOptionsEx.ImplementationID is valid
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Write implemetation ID.

---

## 1.5.762 BS\_UDFOPT\_PARTITION\_TYPE Macro

**C++**

```
#define BS_UDFOPT_PARTITION_TYPE 0x00000002      // SUDFOptionsEx.PartitionType is valid
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Write partition type.

---

## 1.5.763 BS\_UDFOPT\_VERSION Macro

**C++**

```
#define BS_UDFOPT_VERSION 0x00000001      // SUDFOptionsEx.Version is valid
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

UDF options: Write UDF version.

---

## 1.5.764 BS\_UNDERRUN\_PROTECTION Macro

**C++**

```
#define BS_UNDERRUN_PROTECTION 0x002000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to perform buffer protection functions like Burn-Proof.

---

## 1.5.765 BS\_UNKNOWN Macro

**C++**

```
#define BS_UNKNOWN 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

A disc identifier for disc type: Unknown

---

## 1.5.766 BS\_UPC\_READ Macro

**C++**

```
#define BS_UPC_READ 0xFFFF00000000DLL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to read the UPC info of a disc.

---

## 1.5.767 BS\_VCD\_INFINITE Macro

**C++**

```
#define BS_VCD_INFINITE 0xFFFF
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_INFINITE.

---

## 1.5.768 BS\_VCD\_KEY\_0 Macro

**C++**

```
#define BS_VCD_KEY_0 0
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_KEY\_0.

---

## 1.5.769 BS\_VCD\_KEY\_DEFAULT Macro

**C++**

```
#define BS_VCD_KEY_DEFAULT 101
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_KEY\_DEFAULT.

---

## 1.5.770 BS\_VCD\_KEY\_NEXT Macro

**C++**

```
#define BS_VCD_KEY_NEXT 104
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_KEY\_NEXT.

---

## 1.5.771 BS\_VCD\_KEY\_PREVIOUS Macro

**C++**

```
#define BS_VCD_KEY_PREVIOUS 103
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_KEY\_PREVIOUS.

---

## 1.5.772 BS\_VCD\_KEY\_RETURN Macro

**C++**

```
#define BS_VCD_KEY_RETURN 102
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_KEY\_RETURN.

---

## 1.5.773 BS\_VCD\_SEGMENT\_ITEM\_0 Macro

**C++**

```
#define BS_VCD_SEGMENT_ITEM_0 1000
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro BS\_VCD\_SEGMENT\_ITEM\_0.

## 1.5.774 BS\_VCD\_TRACK\_ITEM\_0 Macro

**C++**

```
#define BS_VCD_TRACK_ITEM_0 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro BS\_VCD\_TRACK\_ITEM\_0.

---

## 1.5.775 BS\_WHOLE\_PATH Macro

**C++**

```
#define BS_WHOLE_PATH 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

The complete path of the file is created as a subfolder in the chDestinationPath of the project.

---

## 1.5.776 BS\_WM.DAO Macro

**C++**

```
#define BS_WM.DAO 1
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Write method: DAO (Disc-At-Once)

---

## 1.5.777 BS\_WM.DAO96 Macro

**C++**

```
#define BS_WM.DAO96 2
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Write method: DAO96 (Disc-At-Once + 96)

## 1.5.778 BS\_WM\_TAO Macro

**C++**

```
#define BS_WM_TAO 0
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Write method: TAO (Track-At-Once)

---

## 1.5.779 BS\_WRITE\_BLURAY\_R Macro

**C++**

```
#define BS_WRITE_BLURAY_R 0x000010000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write Blu-Ray Recordable.

---

## 1.5.780 BS\_WRITE\_BLURAY\_R\_XL Macro

**C++**

```
#define BS_WRITE_BLURAY_R_XL 0xFFFFF00000025LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write BD-R XL.

---

## 1.5.781 BS\_WRITE\_BLURAY\_RE Macro

**C++**

```
#define BS_WRITE_BLURAY_RE 0x000020000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write Blu-Ray ReWritable.

---

## 1.5.782 BS\_WRITE\_BLURAY\_RE\_XL Macro

**C++**

```
#define BS_WRITE_BLURAY_RE_XL 0xFFFFF00000026LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write BD-RE XL (Re-Writeable).

---

## 1.5.783 BS\_WRITE\_CDR Macro

**C++**

```
#define BS_WRITE_CDR 0x000000080000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write CD-Recordable.

---

## 1.5.784 BS\_WRITE\_CDRW Macro

**C++**

```
#define BS_WRITE_CDRW 0x000000100000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write CD-ReWritable.

---

## 1.5.785 BS\_WRITE\_CDRW\_CAV Macro

**C++**

```
#define BS_WRITE_CDRW_CAV 0x008000000000LL
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

Drive feature: Device is able to write CD-RW media that is designed for CAV recording.

## 1.5.786 BS\_WRITE\_DVD\_DL Macro

**C++**

```
#define BS_WRITE_DVD_DL 0x000004000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD-R Double Layer.

---

## 1.5.787 BS\_WRITE\_DVD\_MRDL Macro

**C++**

```
#define BS_WRITE_DVD_MRDL 0x000008000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD-R Double Layer.

---

## 1.5.788 BS\_WRITE\_DVD\_RDL\_PLUS Macro

**C++**

```
#define BS_WRITE_DVD_RDL_PLUS 0x200000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD+R Double Layer.

---

## 1.5.789 BS\_WRITE\_DVD\_RWDL\_PLUS Macro

**C++**

```
#define BS_WRITE_DVD_RWDL_PLUS 0x400000000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD+ReWritable Double Layer.

## 1.5.790 BS\_WRITE\_DVDR Macro

**C++**

```
#define BS_WRITE_DVDR 0x000000200000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD-R.

---

## 1.5.791 BS\_WRITE\_DVDR\_PLUS Macro

**C++**

```
#define BS_WRITE_DVDR_PLUS 0x000001000000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD+R.

---

## 1.5.792 BS\_WRITE\_DVDRAM Macro

**C++**

```
#define BS_WRITE_DVDRAM 0x000000800000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD-RAM.

---

## 1.5.793 BS\_WRITE\_DVDRW Macro

**C++**

```
#define BS_WRITE_DVDRW 0x000000400000LL
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

Drive feature: Device is able to write DVD-ReWritable.

## 1.5.794 BS\_WRITE\_DVDRW\_PLUS Macro

### C++

```
#define BS_WRITE_DVDRW_PLUS 0x000002000000LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device is able to write DVD+ReWritable.

---

## 1.5.795 BS\_WRITE\_HDDVD\_R Macro

### C++

```
#define BS_WRITE_HDDVD_R 0x000040000000LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device is able to write HDDVD Reordable.

---

## 1.5.796 BS\_WRITE\_HDDVD\_RW Macro

### C++

```
#define BS_WRITE_HDDVD_RW 0x000080000000LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device is able to write HDDVD ReWritable.

---

## 1.5.797 BS\_WRITE\_MOUNT\_RAINER Macro

### C++

```
#define BS_WRITE_MOUNT_RAINER 0x000400000000LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device is able to write Mount Rainer.

## 1.5.798 BS\_WRITE\_TEST Macro

### C++

```
#define BS_WRITE_TEST 0x001000000000LL
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

Drive feature: Device is able to simulate burning.

### Notes

Simulate burning is a CD-R(W) feature. Not available for DVD or BD.

---

## 1.5.799 COMPRESS\_BLOCKSIZE Macro

### C++

```
#define COMPRESS_BLOCKSIZE 8192
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

This is macro COMPRESS\_BLOCKSIZE.

---

## 1.5.800 ENCRYPT\_BLOCKSIZE Macro

### C++

```
#define ENCRYPT_BLOCKSIZE 2048
```

### File

IsoSDKDefinitions.h (see page 517)

### Description

This is macro ENCRYPT\_BLOCKSIZE.

---

## 1.5.801 FOpen Macro

### C++

```
#define FOpen fopen
```

### File

IsoSDKUnicode.h (see page 587)

### Description

This is macro FOpen.

## 1.5.802 INSERT\_STRUCTURE\_PADDING Macro

### C++

```
#define INSERT_STRUCTURE_PADDING(nBytes) \
    int8 x__the_private_padding[ (nBytes)];
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro INSERT\_STRUCTURE\_PADDING.

---

## 1.5.803 int2bool Macro

### C++

```
#define int2bool(nTmp) (0 != nTmp)
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro int2bool.

---

## 1.5.804 int2BS\_BOOL Macro

### C++

```
#define int2BS_BOOL(nTmp) ((0 != nTmp) ? BS_TRUE : BS_FALSE)
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro int2BS\_BOOL.

---

## 1.5.805 PATHSEPSTRING Macro

### C++

```
#define PATHSEPSTRING " / "
```

### File

IsoSDKDefinitions.h ([see page 517](#))

### Description

This is macro PATHSEPSTRING.

## 1.5.806 SPrintf Macro

**C++**

```
#define SPrintf sprintf
```

**File**

IsoSDKUnicode.h ( see page 587)

**Description**

This is macro SPrintf.

---

## 1.5.807 StrCmp Macro

**C++**

```
#define StrCmp strcmp
```

**File**

IsoSDKUnicode.h ( see page 587)

**Description**

This is macro StrCmp.

---

## 1.5.808 StrCpy Macro

**C++**

```
#define StrCpy strcpy
```

**File**

IsoSDKUnicode.h ( see page 587)

**Description**

This is macro StrCpy.

---

## 1.5.809 StrLen Macro

**C++**

```
#define StrLen strlen
```

**File**

IsoSDKUnicode.h ( see page 587)

**Description**

This is macro StrLen.

## 1.5.810 StrnSet Macro

**C++**

```
#define StrnSet strnset
```

**File**

IsoSDKUnicode.h ([see page 587](#))

**Description**

This is macro StrnSet.

---

## 1.5.811 STRUCTURE\_PADDING\_BIG Macro

**C++**

```
#define STRUCTURE_PADDING_BIG (3*sizeof(void*) + 16)
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro STRUCTURE\_PADDING\_BIG.

---

## 1.5.812 STRUCTURE\_PADDING\_HUGE Macro

**C++**

```
#define STRUCTURE_PADDING_HUGE (4*sizeof(void*) + 32 * sizeof(TCHAR) )
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro STRUCTURE\_PADDING\_HUGE.

---

## 1.5.813 STRUCTURE\_PADDING\_NORMAL Macro

**C++**

```
#define STRUCTURE_PADDING_NORMAL (2*sizeof(void*) + 8)
```

**File**

IsoSDKDefinitions.h ([see page 517](#))

**Description**

This is macro STRUCTURE\_PADDING\_NORMAL.

---

## 1.5.814 STRUCTURE\_PADDING\_SMALL Macro

**C++**

```
#define STRUCTURE_PADDING_SMALL (sizeof(void*) + 4)
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro STRUCTURE\_PADDING\_SMALL.

---

## 1.5.815 STRUCTURE\_PADDING\_STRING\_DATA Macro

**C++**

```
#define STRUCTURE_PADDING_STRING_DATA ( (4*sizeof(void*) + 32) + 2 * (_MAX_PATH) * sizeof(TCHAR) )
```

**File**

IsoSDKDefinitions.h (see page 517)

**Description**

This is macro STRUCTURE\_PADDING\_STRING\_DATA.

---

## 1.5.816 VsPrintf Macro

**C++**

```
#define VsPrintf vsprintf
```

**File**

IsoSDKUnicode.h (see page 587)

**Description**

This is macro VsPrintf.

---

## 1.6 Files

The following table lists files in this documentation.

**Files**

Name	Description
IsoSDKBurningLib.h (see page 517)	This is file IsoSDKBurningLib.h.
IsoSDKDefinitions.h (see page 517)	This is file IsoSDKDefinitions.h.
IsoSDKExport.h (see page 577)	This is file IsoSDKExport.h.
IsoSDKUnicode.h (see page 587)	This is file IsoSDKUnicode.h.

## 1.6.1 IsoSDKBurningLib.h

This is file IsoSDKBurningLib.h.

### Macros

Name	Description
<code>__BS_LIBRARY_H__</code> (see page 305)	This is macro <code>__BS_LIBRARY_H__</code> .
<code>BS_API</code> (see page 308)	VideoCD / SuperVideoCD - MPEG file has a 16:9 display

## 1.6.2 IsoSDKDefinitions.h

This is file IsoSDKDefinitions.h.

### Macros

Name	Description
<code>stat64</code> (see page 305)	This is macro <code>stat64</code> .
<code>__ISOSDK_DEFINITIONS_H__</code> (see page 306)	This is macro <code>__ISOSDK_DEFINITIONS_H__</code> .
<code>_max</code> (see page 306)	This is macro <code>_max</code> .
<code>_min</code> (see page 306)	This is macro <code>_min</code> .
<code>_MAX_PATH</code> (see page 307)	This is macro <code>_MAX_PATH</code> .
<code>BOOL2bool</code> (see page 307)	This is macro <code>BOOL2bool</code> .
<code>BS_ANALOG_AUDIO_PLAYBACK</code> (see page 308)	Drive feature: Device support playback analog audio.
<code>BS_AR_16TO9_DISPLAY</code> (see page 308)	Video CD / Super Video CD - MPEG file has a 16:9 display.
<code>BS_AR_221TO2_DISPLAY</code> (see page 308)	Video CD / Super Video CD - MPEG file has a 2.21:1 display.
<code>BS_AR_4TO3_DISPLAY</code> (see page 309)	Video CD / Super Video CD - The MPEG file has a 4:3 display.
<code>BS_AR_SQUARE_PIXELS</code> (see page 309)	Video CD / Super Video CD - MPEG file has square pixels aspect ratio.

	BS_AR_UNKNOWN (see page 309)	Video CD / Super Video CD - MPEG file has an unknown aspect ratio.
	BS_ASPI_FROGASPI (see page 309)	The FrogAspi DLL. This is a very good replacement of the Adaptec interface. It supports all new interfaces.
	BS_ASPI_INTERNAL (see page 310)	Use the IsoSDK internal ASPI. This is recommended for all Linux and macOS versions and Windows from Windows 2000 and above.
	BS_ASPI_WNASPI (see page 310)	The default external ASPI layer like the one from Adaptec. Please note that this ASPI does neither support USB nor FireWire.
	BS_AUDIO_MP3 (see page 310)	Audio format is MP3 (MPEG Audio Layer 3).
	BS_AUDIO_NO (see page 310)	Audio format is unknown.
	BS_AUDIO_OGG (see page 311)	Audio format is Ogg Vorbis.
	BS_AUDIO_PCM (see page 311)	Audio format is PCM.
	BS_BARCODE_READ (see page 311)	Drive feature: Device is able to read the disc barcode information.
	BS_BLURAY_R (see page 311)	A disc identifier for disc type: Blu-ray BD-R
	BS_BLURAY_R_RRM (see page 312)	A disc identifier for disc type: Blu-ray BD-RRM
	BS_BLURAY_RE (see page 312)	A disc identifier for disc type: Blu-ray BD-RE
	BS_BLURAY_ROM (see page 312)	A disc identifier for disc type: Blu-ray BD-ROM
	BS_BOOL (see page 312)	This is macro BS_BOOL.

	BS_BOOL2bool (see page 313)	Audio format is MP3 (MPEG Audio Layer 3).
	BS_BT_AVERAGE (see page 313)	Audio encoding: Average <a href="#">Bitrate (CBR)</a>
	BS_BT_CONSTANT (see page 313)	Audio encoding: Constant <a href="#">Bitrate (CBR)</a>
	BS_BT_VARIABLE (see page 313)	Audio encoding: Variable <a href="#">Bitrate (VBR)</a>
	BS_BURN_DEVICE (see page 314)	IsoSDK device type: Burn (see page 121) Device.
	BS_BURNER_AUDIO (see page 314)	IsoSDK project type AudioCD.
	BS_BURNER_BLURAY (see page 314)	IsoSDK project type Blu-Ray disc. This project type is for Blu-ray Video. It do not contain a ISO bridge. For ISO/UDF Blu-ray please use BS_BURNER_MIXEDMODE (see page 315) project type.
	BS_BURNER_CUE (see page 314)	IsoSDK project type Bin/Cue Image.
	BS_BURNER_DATA (see page 315)	IsoSDK project type Data CD/DVD. ISO9660 file system + extension.
	BS_BURNER_ISOUDF (see page 315)	IsoSDK project type Data CD/DVD with ISO UDF Bridge.
	BS_BURNER_MIXEDMODE (see page 315)	IsoSDK project type Mixed Mode CD (Audio/Data).
	BS_BURNER_RAW (see page 315)	IsoSDK project type RAW Image (Universal). This project type can burn any type of disc images until you know the details.
	BS_BURNER_SVCD (see page 316)	IsoSDK project type SuperVideoCD.

	BS_BURNER_UDFDVD (see page 316)	IsoSDK project type Data CD/DVD with UDF only.
	BS_BURNER_VCD (see page 316)	IsoSDK project type VideoCD.
	BS_BURNER_VIDEODVD (see page 316)	IsoSDK type VideoDVD (ISO/UDF 1.02). This project type contain folders for VideoDVD and special sorting features for VideoDVD.
	BS_C2_POINTERS (see page 317)	Drive feature: Device support reading C2 error pointers and C2 block error flags.
	BS_CD_R (see page 317)	A disc identifier for disc type: CD-R
	BS_CD_ROM (see page 317)	A disc identifier for disc type: CD-ROM
	BS_CD_RW (see page 318)	A disc identifier for disc type: CD-ReWritable
	BS_CD_TEXT_READ (see page 318)	Drive feature: Device is able to read the CD-TEXT information of a disc (subchannel).
	BS_CD_TEXT_WRITE (see page 318)	Drive feature: Device is able to write the CD-TEXT information of a disc (subchannel).
	BS_CDDA_COMMANDS (see page 318)	Drive feature: Device is able to work with CDDA command set.
	BS_CDDA_STREAM_IS_ACCURATE (see page 319)	Drive feature: The Logical Unit shall support an audio location without losing place to continue the READ CD command.
	BS_CDTCI_ARRANGER (see page 319)	CD-Text content items: Arranger field.
	BS_CDTCI_COMPOSER (see page 319)	CD-Text content items: Composer field.

	BS_CDTCI_MESSAGE (see page 319)	CD-Text content items: Message field.
	BS_CDTCI_PERFORMER (see page 320)	CD-Text content items: Performer field.
	BS_CDTCI_SONG_WRITER (see page 320)	CD-Text content items: Songwriter field.
	BS_CDTCI_TITLE (see page 320)	CD-Text content items: Title field.
	BS_CHANGER_SIDE_CHANGE_CAPABLE (see page 320)	Drive feature: The Logical Unit is capable of selecting both sides of discs.
	BS_CHANGER_SOFTWARE_SLOT_SELECTION (see page 321)	Drive feature: This bit controls the behavior of the LOAD/UNLOAD MEDIUM command when trying to load a Slot with no disc present.
	BS_CHANGER_SUPPORTS_DISC_PRESENT (see page 321)	Drive feature: The Logical Unit contains an embedded changer, and after a reset condition or a magazine change, the Logical Unit is capable of reporting the exact contents of the slots.
	BS_COMPOSITE_AUDIO_AND_VIDEO (see page 321)	Drive feature: The Logical Unit shall be capable of delivering a composite Audio and Video data stream from an independent digital port.
	BS_COMPRESSED_SIGNATURE (see page 321)	This is macro BS_COMPRESSED_SIGNATURE .
	BS_COMPRESSED_SIGNATURE_FOR_DRIVER (see page 322)	This is macro BS_COMPRESSED_SIGNATURE_FOR_DRIVER.

	BS_CONTINUE_LAST_SESSION (see page 322)	Default value to import last session of the inserted disc.
	BS_CONTINUE_NO_SESSION (see page 322)	Default value to do not import any available session.
	BS_CPRMAUTH (see page 322)	Device is able to CPRM authentification and key management.
	BS_CURRENT_DEVICE (see page 323)	Default device index of the current selected drive.
	BS_DAO_16 (see page 323)	Drive feature: Device can write DAO16 method.
	BS_DAO_96_PACK (see page 323)	Drive feature: Device can write DAO96 packet write method.
	BS_DAO_96_RAW (see page 323)	Drive feature: Device can write DAO96 raw write method.
	BS_DAO_RAW (see page 324)	Drive feature: Device can write DAO raw write method.
	BS_DD_CD_R (see page 324)	A disc identifier for disc type: Double Density Compact Disc - Recording
	BS_DD_CD_ROM (see page 324)	A disc identifier for disc type: Double Density Compact Disc - ROM
	BS_DD_CD_RW (see page 324)	A disc identifier for disc type: Double Density Compact Disc - ReWritable
	BS_DEFAULT_LANGUAGE_FILE (see page 325)	The defined default language file name: "language.ini".
	BS_DEFAULT_REGISTRY_KEY (see page 325)	The defined default registry key: "HKEY_CURRENT_USER".

	BS_DEFAULT_REGISTRY_PATH (see page 325)	The default defined registry path: "\Software\IsoSDK".
	BS_DEFECTMANAGEMENT (see page 325)	Drive feature: Device has defect management feature. Available to provide defect-free address space. For example BD recorders provide this feature.
	BS_DIGITAL_PORT_1 (see page 326)	Drive feature: The Logical Unit shall support digital output (IEC958) on port 1.
	BS_DIGITAL_PORT_2 (see page 326)	Drive feature: The Logical Unit shall support digital output (IEC958) on port 2.
	BS_DONT_SAVE_PATH (see page 326)	The file is created in the chDestinationPath of the project.
	BS_DVD_MRDL (see page 326)	A disc identifier for disc type: DVD-R Double Layer
	BS_DVD_PLUSR (see page 327)	A disc identifier for disc type: DVD+R
	BS_DVD_PLUSRW (see page 327)	A disc identifier for disc type: DVD+R ReWritable
	BS_DVD_R (see page 327)	A disc identifier for disc type: DVD-R
	BS_DVD_RAM (see page 327)	A disc identifier for disc type: DVD-RAM
	BS_DVD_RDL_PLUS (see page 328)	A disc identifier for disc type: DVD+R Doube Layer
	BS_DVD_ROM (see page 328)	A disc identifier for disc type: DVD-ROM

	BS_DVD_RW (see page 328)	A disc identifier for disc type: DVD-R ReWriteable
	BS_DVD_RW_RO (see page 328)	A disc identifier for disc type: DVD-R ReWriteable - Restricted Overwrite
	BS_DVD_RW_SR (see page 329)	A disc identifier for disc type: DVD-R ReWriteable - Sequential Recording
	BS_DVD_RWDL_PLUS (see page 329)	A disc identifier for disc type: DVD+R ReWriteable Double Layer
	BS_EJECT_INDIVIDUAL_OR_MAGAZINE (see page 329)	Drive feature: The Logical Unit shall support media eject via the START STOP UNIT command with the LoEj bit set.
	BS_EMT_BD (see page 329)	Medium type: Blu-Ray medium
	BS_EMT_CD_AUDIO (see page 330)	Medium type: CD-DA format
	BS_EMT_CD_ENHANCED (see page 330)	Medium type: CD-Enhanced format
	BS_EMT_CD_MIXED_MODE (see page 330)	Medium type: Mixed Mode CD format
	BS_EMT_CD_MULTISESSION (see page 330)	Medium type: Multi-session data CD
	BS_EMT_CD_ROM (see page 331)	Medium type: Data CD or CD of unknown format
	BS_EMT_CD_ROM_XA (see page 331)	Medium type: CD-eXtended Architecture format
	BS_EMT_DVD (see page 331)	Medium type: DVD medium
	BS_EMT_HDDVD (see page 331)	Medium type: HDDVD medium
	BS_ET_AAC (see page 332)	Encoding type: AAC (Advanced Audio Codec)

	BS_ET_FLAC (see page 332)	Encoding type: FLAC ( <a href="#">Free Lossless Audio Codec</a> ).
	BS_ET_MP3 (see page 332)	Encoding type: MP3 ( <a href="#">MPEG Audio Layer 3</a> ).
	BS_ET_MP4 (see page 332)	Encoding type: MEPG-4 ( <a href="#">Mp4</a> ).
	BS_ET_OGG (see page 333)	Encoding type: <a href="#">Ogg Vorbis</a>
	BS_ET_OPUS (see page 333)	Encoding type: <a href="#">Opus</a>
	BS_ET_WMA (see page 333)	Encoding type: WMA (Windows Media Audio)
	BS_FA_ADVANCED_HIDDEN (see page 333)	File attribute: Advanced hidden files. Not valid in all OS. Use in combination with ISO files.
	BS_FA_ALL (see page 334)	File attribute: All attributes set.
	BS_FA_ARCHIVE (see page 334)	File attribute: Archive file.
	BS_FA_DIRECTORY (see page 334)	File attribute: Folder / Directory.
	BS_FA_HIDDEN (see page 334)	File attribute: Hidden file.
	BS_FA_READONLY (see page 335)	File attribute: Write protected files.
	BS_FA_SYSTEM (see page 335)	File attribute: System files.
	BS_FALSE (see page 335)	IsoSDK BOOL value: FALSE
	BS_FF_BIN (see page 335)	File format is BIN (2352).
	BS_FF_ISO (see page 336)	File format is ISO (2048)
	BS_FF_MPEG (see page 336)	File format is MPEG.
	BS_FF_WAVE (see page 336)	File format is WAVE (PCM).
	BS_FS_BOOTABLE (see page 336)	Disc file system: El Torito bootable extension to ISO 9660.
	BS_FS_ISO9660 (see page 337)	Disc file system: ISO 9660.
	BS_FS_JOLIET (see page 337)	Disc file system: <a href="#">Joliet</a> extension to ISO 9660.

	BS_FS_ROCKRIDGE (see page 337)	Disc file system: Rock Ridge.
	BS_FS_UDF (see page 337)	Disc file system: Universal Disk Format.
	BS_FS_UNKNOWN (see page 338)	Disc file system: No file system or unknown file system.
	BS_GUI_ERROR_01 (see page 338)	GUI error string: Password and confirmation do not match.
	BS_GUI_ERROR_02 (see page 338)	GUI error string: Password is empty.
	BS_GUI_RESOURCE_01 (see page 338)	GUI resource string: CD/DVD Eraser (Title of erase dialog)
	BS_GUI_RESOURCE_02 (see page 339)	GUI resource string: CD/DVD Burner (Title of burn window).
	BS_GUI_RESOURCE_03 (see page 339)	GUI resource string: Device
	BS_GUI_RESOURCE_04 (see page 339)	GUI resource string: Progress
	BS_GUI_RESOURCE_05 (see page 339)	GUI resource string: Status
	BS_GUI_RESOURCE_06 (see page 340)	GUI resource string: Fast
	BS_GUI_RESOURCE_07 (see page 340)	GUI resource string: Complete
	BS_GUI_RESOURCE_08 (see page 340)	GUI resource string: Exit
	BS_GUI_RESOURCE_09 (see page 340)	GUI resource string: Settings
	BS_GUI_RESOURCE_10 (see page 341)	GUI resource string: Start
	BS_GUI_RESOURCE_11 (see page 341)	GUI resource string: Stop
	BS_GUI_RESOURCE_12 (see page 341)	GUI resource string: Simulate burning
	BS_GUI_RESOURCE_13 (see page 342)	GUI resource string: Finalize medium
	BS_GUI_RESOURCE_14 (see page 342)	GUI resource string: Finalize medium
	BS_GUI_RESOURCE_15 (see page 342)	GUI resource string: Speed
	BS_GUI_RESOURCE_16 (see page 342)	GUI resource string: Eject medium after burn

	BS_GUI_RESOURCE_17 (see page 343)	GUI resource string: Joliet file system
	BS_GUI_RESOURCE_18 (see page 343)	GUI resource string: Cache size
	BS_GUI_RESOURCE_19 (see page 343)	GUI resource string: 0.5 MB
	BS_GUI_RESOURCE_20 (see page 343)	GUI resource string: 64 MB
	BS_GUI_RESOURCE_21 (see page 344)	GUI resource string: Medium name
	BS_GUI_RESOURCE_22 (see page 344)	GUI resource string: Make boot medium
	BS_GUI_RESOURCE_23 (see page 344)	GUI resource string: No of Copies
	BS_GUI_RESOURCE_24 (see page 345)	GUI resource string: Medium
	BS_GUI_RESOURCE_25 (see page 345)	GUI resource string: Burning
	BS_GUI_RESOURCE_26 (see page 345)	GUI resource string: General
	BS_GUI_RESOURCE_27 (see page 345)	GUI resource string: Simulation was successful
	BS_GUI_RESOURCE_28 (see page 346)	GUI resource string: Burn (see page 121) the medium now?
	BS_GUI_RESOURCE_29 (see page 346)	GUI resource string: Verify data after burn
	BS_GUI_RESOURCE_30 (see page 346)	GUI resource string: DVD High-Compatibility-Mode
	BS_GUI_RESOURCE_31 (see page 346)	GUI resource string: Bytes Written:
	BS_GUI_RESOURCE_32 (see page 347)	GUI resource string: Elapsed Time:
	BS_GUI_RESOURCE_33 (see page 347)	GUI resource string: Buffer:
	BS_GUI_RESOURCE_34 (see page 347)	GUI resource string: Remaining Time:
	BS_GUI_RESOURCE_35 (see page 347)	GUI resource string: Other
	BS_GUI_RESOURCE_36 (see page 348)	GUI resource string: Compression
	BS_GUI_RESOURCE_37 (see page 348)	GUI resource string: Encryption

	BS_GUI_RESOURCE_38 (see page 348)	GUI resource string: Password
	BS_GUI_RESOURCE_39 (see page 349)	GUI resource string: UDF
	BS_GUI_RESOURCE_40 (see page 349)	GUI resource string: Version
	BS_GUI_RESOURCE_41 (see page 349)	GUI resource string: Partition type
	BS_GUI_RESOURCE_42 (see page 349)	GUI resource string: Save Log
	BS_GUI_RESOURCE_43 (see page 350)	GUI resource string: Select multiple devices
	BS_GUI_RESOURCE_44 (see page 350)	GUI resource string: Multiple devices
	BS_GUI_RESOURCE_45 (see page 350)	GUI resource string: Write method
	BS_GUI_RESOURCE_46 (see page 350)	GUI resource string: Confirm
	BS_GUI_RESOURCE_47 (see page 351)	GUI resource string: Write streams
	BS_GUI_RESOURCE_48 (see page 351)	GUI resource string: Rescan
	BS_GUI_RESOURCE_49 (see page 351)	GUI resource string: Auto Erase (see page 153)
	BS_GUI_RESOURCE_50 (see page 352)	GUI resource string: Rock Ridge File System
	BS_GUI_RESOURCE_51 (see page 352)	GUI resource string: Pad Data Tracks
	BS_GUI_RESOURCE_52 (see page 352)	GUI resource string: Eject (Erase (see page 153) Dialog)
	BS_GUI_RESOURCE_53 (see page 352)	GUI resource string: AVCHD disc
	BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLUMN (see page 353)	GUI resource string: Artist
	BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_COLUMN (see page 353)	GUI resource string: Category
	BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLUMN (see page 353)	GUI resource string: DisclD
	BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABEL (see page 354)	GUI resource string: Status
	BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE (see page 354)	GUI resource string: Title
	BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN (see page 354)	GUI resource string: Title

	BS_HD_DVD_R ( <a href="#">see page 355</a> )	A disc identifier for disc type: HDDVD R
	BS_HD_DVD_R_DL ( <a href="#">see page 355</a> )	A disc identifier for disc type: HDDVD R Double Layer
	BS_HD_DVD_RAM ( <a href="#">see page 355</a> )	A disc identifier for disc type: HDDVD-RAM
	BS_HD_DVD_ROM ( <a href="#">see page 355</a> )	A disc identifier for disc type: HDDVD-ROM
	BS_HD_DVD_RW ( <a href="#">see page 356</a> )	A disc identifier for disc type: HDDVD-ReWriteable
	BS_HD_DVD_RW_DL ( <a href="#">see page 356</a> )	A disc identifier for disc type: HDDVD ReWriteable Double Layer
	BS_IL_HIGH_DEBUG ( <a href="#">see page 356</a> )	Log info level: High. Use only for debugging.
	BS_IL_INFO ( <a href="#">see page 356</a> )	Log info level: Info. Use this for your own log files.
	BS_IL_LOW_DEBUG ( <a href="#">see page 357</a> )	Log info level: Low. Use only for debugging.
	BS_IL_MEDIUM_DEBUG ( <a href="#">see page 357</a> )	Log info level: Medium. Use only for debugging.
	BS_ILE_TOO_BIG_FILE ( <a href="#">see page 357</a> )	ISO level error: File is to Big.
	BS_ILE_TOO_LONG_DIRECTORY_NESTING ( <a href="#">see page 357</a> )	ISO level error: To long Directory
	BS_IMG_BIN ( <a href="#">see page 358</a> )	A image type BIN (RAW) with a blocksize of 2352.
	BS_IMG_ISO ( <a href="#">see page 358</a> )	A image type ISO with a blocksize of 2048.
	BS_IMGTASK_CREATE ( <a href="#">see page 358</a> )	The IsoSDK will create a image file.
	BS_IMGTASK_VERIFY ( <a href="#">see page 358</a> )	The IsoSDK will verify a image file.
	BS_IMPOPTS_COMMON ( <a href="#">see page 359</a> )	This is macro BS_IMPOPTS_C COMMON.

	BS_IMPOPTS_DECRYPT (see page 359)	This is macro BS_IMPOPTS_D ECRYPT.
	BS_IMPOPTS_EX (see page 359)	This is macro BS_IMPOPTS_E X.
	BS_IMPOPTS_UNCOMPRESS (see page 359)	This is macro BS_IMPOPTS_U NCOMPRESS.
	BS_INVALID_HANDLE (see page 360)	IsoSDK error code: The submitted handle is not valid or not initialized.
	BS_INVALID_TAG_HANDLE (see page 360)	IsoSDK error code: A created but not initialized Handle to receive Tags of a disc.
	BS_ISO_LEVEL_1 (see page 360)	<p>Disc file system: ISO Level 1 Specification:</p> <ol style="list-style-type: none"> <li>1. Name format is 8.3. Available chars: only capital letters and underscore.</li> <li>2. File extension length is limited to 3 chars.</li> <li>3. Directory can't has extension.</li> <li>4. File size limit is 2 GB.</li> </ol>
	BS_ISO_LEVEL_2 (see page 360)	<p>Disc file system: ISO Level 2. Specification:</p> <ol style="list-style-type: none"> <li>1. Max name length is 31 chars.</li> <li>2. File size limit is 2 GB.</li> </ol>

	BS_ISO_LEVEL_3 (see page 361)	Disc file system: ISO Level 3. Specification: 1. Max name length is 128 chars. 2. No file size limit.
	BS_ISO_LEVEL_ROMEO (see page 361)	Disc file system: ISO Level Romeo. Specification: 1. Max name length is 128 chars. 2. Lower case chars are allowed in file names.
	BS_ISRC_READ (see page 361)	Drive feature: Device is able to read the ISRC (International Standard Recording Code) info of a disc.
	BS_LABELFLASH (see page 362)	Drive feature: Device is able to write Labelflash discs.
	BS_LAYER_JUMP_RECORDING (see page 362)	Drive feature: Device is able to perform Layer Jump Recording.
	BS_LIGHTSCRIBE (see page 362)	Drive feature: Device is able to write Lightscribe discs.
	BS_LOCK_MEDIA (see page 362)	Drive feature: Device is able to lock the media.
	BS_LOCK_STATE (see page 363)	Drive feature: Device is able return the lock state.
	BS_LP_BOOL (see page 363)	This is macro BS_LP_BOOL.
	BS_LS_COMPLETE_SESSION (see page 363)	Disc session status: Complete.
	BS_LS_DAMAGED_SESSION (see page 363)	Disc session status: Session is damaged.
	BS_LS_EMPTY_SESSION (see page 364)	Disc session status: Session is empty.

	BS_LS_INCOMPLETE_SESSION (see page 364)	Disc session status: Session is incomplete.
	BS_MAX_SPEED (see page 364)	This represents the max available speed of the device.
	BS_METHOD_2_ADDRESSING_FIXED_PACKETS (see page 364)	Drive feature: Device support reading fixed packet tracks on CD-R/RW media where the Addressing type is Method 2.
	BS_MODE2_FORM1_READ (see page 365)	Drive feature: Device is able to read disc that are recorded with Mode2 Form1 format.
	BS_MODE2_FORM2_READ (see page 365)	Drive feature: Device is able to read disc that are recorded with Mode2 Form2 format.
	BS_MS_COMPLETE_DISK (see page 365)	Disc status: Disc is complete / finalized.
	BS_MS_EMPTY_DISK (see page 365)	Disc status: Disc is empty
	BS_MS_INCOMPLETE_DISK (see page 366)	Disc status: Disc is incomplete (not finalized).
	BS_MS_OTHER (see page 366)	Disc status: Unknown
	BS_MULTI_ERROR_01 (see page 366)	IsoSDK error code: The disc are not identical.
	BS_MULTI_ERROR_02 (see page 366)	IsoSDK error code: Invalid write speed.
	BS_MULTISESSION (see page 367)	Drive feature: Device is able to perform multi-session disc.
	BS_NDEF (see page 367)	This is macro BS_NDEF.
	BS_PACKET_WRITE (see page 367)	Drive feature: The drive can perform packet writing.

	BS_PARENTDIR_ONLY (see page 367)	Only the superordinate directory of the file will be created in chDestinationPath of the project. Example: chSourceDir = c:\tmp1\tmp2\test.tmp - only the subfolder tmp2 will be created.
	BS_PREVENT_JUMPER (see page 368)	Drive feature: The Logical Unit has a physical jumper named the Prevent/Allow Jumper and the jumper is present.
	BS_R_W_SUBCHANNELS_DEINT_AND_CORR (see page 368)	Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command with the returned data de-interleaved and error corrected
	BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ (see page 368)	Drive feature: The Logical Unit is capable of reading the raw R-W Sub-channel information from the Lead-in.
	BS_R_W_SUBCHANNELS_READ (see page 368)	Drive feature: The Logical Unit shall support reading R-W sub-channel via the READ CD command.
	BS_RDT_DATA (see page 369)	Input data contains UserData field
	BS_RDT_EDC_ECC (see page 369)	Input data contains ECC ( <a href="#">Error Correction Code</a> ) /EDC (Error Detection Code) fields

	BS_RDT_SUBCH_PQ (see page 369)	Input data contains P-Q subchannel data in packed form
	BS_RDT_SUBCH_PW (see page 369)	Input data contains P-W subchannel data in raw form.
	BS_RDT_SUBCH_RW (see page 370)	Input data contains R-W subchannel data in raw form.
	BS_RDT_SUBHEADERS (see page 370)	Input data contains Subheaders field.
	BS_RDT_SYNC_HEADER (see page 370)	Input data contains Sync/Header field.
	BS_READ_BLURAY_R (see page 370)	Drive feature: Device is able to read Blu-Ray R.
	BS_READ_BLURAY_R_XL (see page 371)	Drive feature: Device is able to read BD-R XL.
	BS_READ_BLURAY_RE (see page 371)	Drive feature: Device is able to read Blu-Ray ReWritable.
	BS_READ_BLURAY_RE_XL (see page 371)	Drive feature: Device is able to read BD-RE XL (Re-Writable).
	BS_READ_BLURAY_ROM (see page 371)	Drive feature: Device is able to read Blu-Ray ROM.
	BS_READ_CDR (see page 372)	Drive feature: Device is able to read CD-R.
	BS_READ_CDRW (see page 372)	Drive feature: Device is able to read CD ReWritable.
	BS_READ_CDRW_CAV (see page 372)	Drive feature: Device is able to read CD-RW media that is designed for CAV recording.
	BS_READ_DEVICE (see page 372)	IsoSDK device type: Read Device
	BS_READ_DVD (see page 373)	Drive feature: Device is able to read DVD-R.
	BS_READ_DVD_DL (see page 373)	Drive feature: Device is able to read DVD-R Double Layer.

	BS_READ_DVD_MRDL (see page 373)	Drive feature: Device is able to read DVD-R Double Layer.
	BS_READ_DVD_RDL_PLUS (see page 373)	Drive feature: Device is able to read DVD+R Double Layer.
	BS_READ_DVD_RWDL_PLUS (see page 374)	Drive feature: Device is able to read DVD+ReWritable Double Layer.
	BS_READ_DVDR (see page 374)	Drive feature: Device is able to read DVD-R.
	BS_READ_DVDR_PLUS (see page 374)	Drive feature: Device is able to read DVD+R.
	BS_READ_DVDRAM (see page 374)	Drive feature: Device is able to read DVD-RAM.
	BS_READ_DVDRW (see page 375)	Drive feature: Device is able to read DVD-ReWritable.
	BS_READ_DVDRW_PLUS (see page 375)	Drive feature: Device is able to read DVD+ReWritable.
	BS_READ_HDDVD_R (see page 375)	Drive feature: Device is able to read HDDVD R.
	BS_READ_HDDVD_ROM (see page 375)	Drive feature: Device is able to read HDDVD Rom.
	BS_READ_HDDVD_RW (see page 376)	Drive feature: Device is able to read HDDVD ReWritable.
	BS_READ_MOUNT_RAINER (see page 376)	Drive feature: Device is able to read Mount Rainer.
	BS_RM_RAW (see page 376)	IsoSDK read mode: RAW (2352)
	BS_RM_RAW_SUBCHANNEL (see page 376)	IsoSDK read mode: RAW Subchannel (2352 +16)

	BS_RM_USERDATA (see page 377)	IsoSDK read mode: Userdata
	BS_RTF_AUDIO (see page 377)	Track type: Audio track
	BS_RTF_MODE1 (see page 377)	Track type: Mode 1 track
	BS_RTF_MODE2_FORM1 (see page 377)	Track type: Mode 2 Form 1 track
	BS_RTF_MODE2_FORM2 (see page 378)	Track type: Mode 2 Form 2 track
	BS_RTF_MODE2_FORMLESS (see page 378)	Track type: Mode 2 formless track
	BS_SCSI_ERROR_001 (see page 378)	IsoSDK error code: Operation in progress.
	BS_SCSI_ERROR_002 (see page 378)	IsoSDK error code: Cleaning requested.
	BS_SCSI_ERROR_004 (see page 379)	IsoSDK error code: No reference position found
	BS_SCSI_ERROR_005 (see page 379)	IsoSDK error code: Multiple peripheral devices selected.
	BS_SCSI_ERROR_006 (see page 379)	IsoSDK error code: Unreachable copy target.
	BS_SCSI_ERROR_007 (see page 379)	IsoSDK error code: Track following error.
	BS_SCSI_ERROR_008 (see page 380)	IsoSDK error code: Tracking servo failure.
	BS_SCSI_ERROR_009 (see page 380)	IsoSDK error code: Focus servo failure
	BS_SCSI_ERROR_01 (see page 380)	IsoSDK error code: No additional sense information
	BS_SCSI_ERROR_010 (see page 380)	IsoSDK error code: Spindle servo failure.
	BS_SCSI_ERROR_011 (see page 381)	IsoSDK error code: Head select fault.
	BS_SCSI_ERROR_012 (see page 381)	IsoSDK error code: Head select fault.
	BS_SCSI_ERROR_013 (see page 381)	IsoSDK error code: I-ec uncorrectable error.

	BS_SCSI_ERROR_014 (see page 381)	IsoSDK error code: Read error - Loss of streaming.
	BS_SCSI_ERROR_015 (see page 382)	IsoSDK error code: Recorded entity not found.
	BS_SCSI_ERROR_016 (see page 382)	IsoSDK error code: Record not found.
	BS_SCSI_ERROR_017 (see page 382)	IsoSDK error code: Random positioning error.
	BS_SCSI_ERROR_018 (see page 382)	IsoSDK error code: Parameter list length error.
	BS_SCSI_ERROR_019 (see page 383)	IsoSDK error code: Synchronous data transfer error.
	BS_SCSI_ERROR_02 (see page 383)	IsoSDK error code: I/O process terminated.
	BS_SCSI_ERROR_020 (see page 383)	IsoSDK error code: Miss compare during verify operation.
	BS_SCSI_ERROR_021 (see page 383)	IsoSDK error code: Invalid command operation code.
	BS_SCSI_ERROR_022 (see page 384)	IsoSDK error code: Logical block address out of range.
	BS_SCSI_ERROR_023 (see page 384)	IsoSDK error code: Invalid element address.
	BS_SCSI_ERROR_024 (see page 384)	IsoSDK error code: Invalid address for write.
	BS_SCSI_ERROR_025 (see page 384)	IsoSDK error code: Invalid release of persistent reservation.
	BS_SCSI_ERROR_026 (see page 385)	IsoSDK error code: Data decryption error.
	BS_SCSI_ERROR_027 (see page 385)	IsoSDK error code: Inline data length exceeded.
	BS_SCSI_ERROR_028 (see page 385)	IsoSDK error code: Invalid operation for copy source or destination.

	BS_SCSI_ERROR_029 (see page 385)	IsoSDK error code: Not ready to ready change, medium may have changed.
	BS_SCSI_ERROR_030 (see page 386)	IsoSDK error code: Import or export element accessed.
	BS_SCSI_ERROR_031 (see page 386)	IsoSDK error code: Power on, reset, or bus device reset occurred.
	BS_SCSI_ERROR_032 (see page 386)	IsoSDK error code: Power on occurred.
	BS_SCSI_ERROR_033 (see page 386)	IsoSDK error code: SCSI bus reset occurred.
	BS_SCSI_ERROR_034 (see page 387)	IsoSDK error code: Bus device reset function occurred.
	BS_SCSI_ERROR_035 (see page 387)	IsoSDK error code: Device internal reset.
	BS_SCSI_ERROR_036 (see page 387)	IsoSDK error code: Transceiver mode changed to single-ended.
	BS_SCSI_ERROR_037 (see page 387)	IsoSDK error code: Transceiver mode changed to LVD.
	BS_SCSI_ERROR_038 (see page 388)	IsoSDK error code: Reservations pre-empted.
	BS_SCSI_ERROR_039 (see page 388)	IsoSDK error code: Reservations released.
	BS_SCSI_ERROR_040 (see page 388)	IsoSDK error code: Registrations pre-empted.
	BS_SCSI_ERROR_041 (see page 388)	IsoSDK error code: Copy cannot execute since host cannot disconnect.
	BS_SCSI_ERROR_042 (see page 389)	IsoSDK error code: Current program area is not empty.

	BS_SCSI_ERROR_043 (see page 389)	IsoSDK error code: Current program area is empty.
	BS_SCSI_ERROR_044 (see page 389)	IsoSDK error code: Persistent prevent conflict.
	BS_SCSI_ERROR_045 (see page 389)	IsoSDK error code: Insufficient time for operation.
	BS_SCSI_ERROR_046 (see page 390)	IsoSDK error code: Zoned formatting failed due to spare linking.
	BS_SCSI_ERROR_047 (see page 390)	IsoSDK error code: Enclosure failure.
	BS_SCSI_ERROR_048 (see page 390)	IsoSDK error code: Enclosure services failure.
	BS_SCSI_ERROR_049 (see page 390)	IsoSDK error code: Unsupported enclosure function.
	BS_SCSI_ERROR_050 (see page 391)	IsoSDK error code: Enclosure services unavailable.
	BS_SCSI_ERROR_051 (see page 391)	IsoSDK error code: Enclosure services transfer failure.
	BS_SCSI_ERROR_052 (see page 391)	IsoSDK error code: Enclosure services transfer refused.
	BS_SCSI_ERROR_053 (see page 391)	IsoSDK error code: Invalid bits in identify message.
	BS_SCSI_ERROR_054 (see page 392)	IsoSDK error code: Target operating conditions have changed.
	BS_SCSI_ERROR_055 (see page 392)	IsoSDK error code: Microcode has been changed.
	BS_SCSI_ERROR_056 (see page 392)	IsoSDK error code: Changed operating definition.
	BS_SCSI_ERROR_057 (see page 392)	IsoSDK error code: Inquiry data has changed.

	BS_SCSI_ERROR_058 (see page 393)	IsoSDK error code: Component device attached.
	BS_SCSI_ERROR_059 (see page 393)	IsoSDK error code: Device identifier change.
	BS_SCSI_ERROR_060 (see page 393)	IsoSDK error code: Redundancy group created or modified.
	BS_SCSI_ERROR_061 (see page 393)	IsoSDK error code: Redundancy group deleted.
	BS_SCSI_ERROR_062 (see page 394)	IsoSDK error code: Spare created or modified.
	BS_SCSI_ERROR_063 (see page 394)	IsoSDK error code: Spare deleted.
	BS_SCSI_ERROR_064 (see page 394)	IsoSDK error code: Reported luns data has changed.
	BS_SCSI_ERROR_065 (see page 394)	IsoSDK error code: Echo buffer overwritten.
	BS_SCSI_ERROR_066 (see page 395)	IsoSDK error code: Message error.
	BS_SCSI_ERROR_067 (see page 395)	IsoSDK error code: Internal target failure.
	BS_SCSI_ERROR_068 (see page 395)	IsoSDK error code: Select or reselect failure.
	BS_SCSI_ERROR_069 (see page 395)	IsoSDK error code: Unsuccessful soft reset.
	BS_SCSI_ERROR_070 (see page 396)	IsoSDK error code: SCSI parity error.
	BS_SCSI_ERROR_071 (see page 396)	IsoSDK error code: Data phase crc error detected.
	BS_SCSI_ERROR_072 (see page 396)	IsoSDK error code: SCSI parity error detected during st data phase.

	BS_SCSI_ERROR_073 (see page 396)	IsoSDK error code: Information unit crc error detected.
	BS_SCSI_ERROR_074 (see page 397)	IsoSDK error code: Asynchronous information protection error detected.
	BS_SCSI_ERROR_075 (see page 397)	IsoSDK error code: Initiator detected error message received.
	BS_SCSI_ERROR_076 (see page 397)	IsoSDK error code: Invalid message error.
	BS_SCSI_ERROR_077 (see page 397)	IsoSDK error code: Data phase error.
	BS_SCSI_ERROR_078 (see page 398)	IsoSDK error code: Insufficient reservation resources.
	BS_SCSI_ERROR_079 (see page 398)	IsoSDK error code: Insufficient resources.
	BS_SCSI_ERROR_080 (see page 398)	IsoSDK error code: Insufficient registration resources.
	BS_SCSI_ERROR_081 (see page 398)	IsoSDK error code: Unable to recover table-of-contents.
	BS_SCSI_ERROR_082 (see page 399)	IsoSDK error code: Operator request or state change input.
	BS_SCSI_ERROR_083 (see page 399)	IsoSDK error code: Operator medium removal request.
	BS_SCSI_ERROR_084 (see page 399)	IsoSDK error code: Operator selected write protect.
	BS_SCSI_ERROR_085 (see page 399)	IsoSDK error code: Operator selected write permit.
	BS_SCSI_ERROR_086 (see page 400)	IsoSDK error code: Threshold condition met.

	BS_SCSI_ERROR_087 (see page 400)	IsoSDK error code: Failure prediction threshold exceeded.
	BS_SCSI_ERROR_088 (see page 400)	IsoSDK error code: Spare area exhaustion prediction threshold exceeded.
	BS_SCSI_ERROR_089 (see page 400)	IsoSDK error code: Failure prediction threshold exceeded (false).
	BS_SCSI_ERROR_090 (see page 401)	IsoSDK error code: Low power condition on.
	BS_SCSI_ERROR_091 (see page 401)	IsoSDK error code: Idle condition activated by timer.
	BS_SCSI_ERROR_092 (see page 401)	IsoSDK error code: Standby condition activated by timer.
	BS_SCSI_ERROR_093 (see page 401)	IsoSDK error code: Idle condition activated by command.
	BS_SCSI_ERROR_094 (see page 402)	IsoSDK error code: Standby condition activated by command.
	BS_SCSI_ERROR_095 (see page 402)	IsoSDK error code: End of user area encountered on this track.
	BS_SCSI_ERROR_096 (see page 402)	IsoSDK error code: Packet does not fit in available space.
	BS_SCSI_ERROR_097 (see page 402)	IsoSDK error code: Illegal mode for this track.
	BS_SCSI_ERROR_098 (see page 403)	IsoSDK error code: Invalid packet size.
	BS_SCSI_ERROR_099 (see page 403)	IsoSDK error code: Voltage fault.

	BS_SCSI_ERROR_100 (see page 403)	IsoSDK error code: Empty or partially written reserved track.
	BS_SCSI_ERROR_101 (see page 403)	IsoSDK error code: No more track reservations allowed.
	BS_SCSI_ERROR_102 (see page 404)	IsoSDK error code: CD control error.
	BS_SCSI_ERROR_103 (see page 404)	IsoSDK error code: Power calibration area almost full.
	BS_SCSI_ERROR_104 (see page 404)	IsoSDK error code: Power calibration area is full.
	BS_SCSI_ERROR_105 (see page 404)	IsoSDK error code: Power calibration area error.
	BS_SCSI_ERROR_106 (see page 405)	IsoSDK error code: RMA/PMA is almost full.
	BS_SCSI_ERROR_107 (see page 405)	IsoSDK error code: Failed to request sense.
	BS_SCSI_ERROR_108 (see page 405)	IsoSDK error code: Medium load or eject failed.
	BS_SCSI_ERROR_109 (see page 405)	IsoSDK error code: Medium removal prevented.
	BS_SCSI_ERROR_110 (see page 406)	IsoSDK error code: Invalid write crossing layer jump.
	BS_SCSI_ERROR_111 (see page 406)	IsoSDK error code: Illegal function.
	BS_SCSI_ERROR_112 (see page 406)	IsoSDK error code: Format-layer may have changed.
	BS_SCSI_ERROR_113 (see page 406)	IsoSDK error code: Enclosure services checksum error.
	BS_SCSI_ERROR_114 (see page 407)	IsoSDK error code: System resource failure.

	BS_SCSI_ERROR_115 (see page 407)	IsoSDK error code: RMZ extention is not allowed.
	BS_SCSI_ERROR_116 (see page 407)	IsoSDK error code: No more test zone extensions are allowed.
	BS_SCSI_ERROR_117 (see page 407)	IsoSDK error code: Current power calibration area is almost full.
	BS_SCSI_ERROR_118 (see page 408)	IsoSDK error code: Current power calibration area is full.
	BS_SCSI_ERROR_119 (see page 408)	IsoSDK error code: RDZ is full.
	BS_SCSI_ERROR_ALLOC_01 (see page 408)	IsoSDK error code: Program memory area update failure.
	BS_SCSI_ERROR_ALLOC_02 (see page 408)	IsoSDK error code: Program memory area is full.
	BS_SCSI_ERROR_ASPI_01 (see page 409)	IsoSDK error code: An ASPI error occurred.
	BS_SCSI_ERROR_ASPI_02 (see page 409)	IsoSDK error code: Invalid host adapter.
	BS_SCSI_ERROR_ASPI_03 (see page 409)	IsoSDK error code: No device has been specified.
	BS_SCSI_ERROR_ASPI_04 (see page 409)	IsoSDK error code: ASPI is busy.
	BS_SCSI_ERROR_ASPI_05 (see page 410)	IsoSDK error code: Bus reset.
	BS_SCSI_ERROR_ASPI_06 (see page 410)	IsoSDK error code: Invalid device ID or device is not a CD/DVD or BD drive.
	BS_SCSI_ERROR_ASPI_07 (see page 410)	IsoSDK error code: The device is busy.
	BS_SCSI_ERROR_ASPI_08 (see page 410)	IsoSDK error code: The target is busy.
	BS_SCSI_ERROR_ASPI_09 (see page 411)	IsoSDK error code: Target reservation conflict.

	BS_SCSI_ERROR_ASPI_10 (see page 411)	IsoSDK error code: The target queue is full.
	BS_SCSI_ERROR_ATT_01 (see page 411)	IsoSDK error code: Warning.
	BS_SCSI_ERROR_ATT_02 (see page 411)	IsoSDK error code: Warning - Specified temperature exceeded.
	BS_SCSI_ERROR_ATT_03 (see page 412)	IsoSDK error code: Warning - Enclosure degraded.
	BS_SCSI_ERROR_ATT_04 (see page 412)	IsoSDK error code: Warning - background self-test failed.
	BS_SCSI_ERROR_ATT_05 (see page 412)	IsoSDK error code: Warning – background pre-scan detected medium error.
	BS_SCSI_ERROR_AUDIO_01 (see page 412)	IsoSDK error code: Audio play operation in progress.
	BS_SCSI_ERROR_AUDIO_02 (see page 413)	IsoSDK error code: Audio play operation paused.
	BS_SCSI_ERROR_AUDIO_03 (see page 413)	IsoSDK error code: Audio play operation successfully completed.
	BS_SCSI_ERROR_AUDIO_04 (see page 413)	IsoSDK error code: Audio play operation stopped due to error.
	BS_SCSI_ERROR_AUDIO_05 (see page 413)	IsoSDK error code: No current audio status to return.
	BS_SCSI_ERROR_CDB_01 (see page 414)	IsoSDK error code: Invalid field in cdb.
	BS_SCSI_ERROR_CDB_02 (see page 414)	IsoSDK error code: CDB decryption error.
	BS_SCSI_ERROR_CIRC_01 (see page 414)	IsoSDK error code: Circ unrecovered error.
	BS_SCSI_ERROR_COMMAND_02 (see page 414)	IsoSDK error code: Command sequence error.

	BS_SCSI_ERROR_COMMAND_03 (see page 415)	IsoSDK error code: Commands cleared by another initiator.
	BS_SCSI_ERROR_COMMAND_04 (see page 415)	IsoSDK error code: Command phase error.
	BS_SCSI_ERROR_COMMAND_05 (see page 415)	IsoSDK error code: Overlapped commands attempted.
	BS_SCSI_ERROR_CRC_01 (see page 415)	IsoSDK error code: De-compression CRC error.
	BS_SCSI_ERROR_DCSS_01 (see page 416)	IsoSDK error code: Copy protection key exchange failure - authentication failure.
	BS_SCSI_ERROR_DCSS_02 (see page 416)	IsoSDK error code: Copy protection key exchange failure - key not present.
	BS_SCSI_ERROR_DCSS_03 (see page 416)	IsoSDK error code: Copy protection key exchange failure - key not established.
	BS_SCSI_ERROR_DCSS_04 (see page 416)	IsoSDK error code: Read of scrambled sector without authentication.
	BS_SCSI_ERROR_DCSS_05 (see page 417)	IsoSDK error code: Medium region code is mismatched to logical unit region.
	BS_SCSI_ERROR_DCSS_06 (see page 417)	IsoSDK error code: Drive region must be permanent/region reset count error.
	BS_SCSI_ERROR_DCSS_07 (see page 417)	IsoSDK error code: Insufficient block count for binding nonce recording.
	BS_SCSI_ERROR_DCSS_08 (see page 417)	IsoSDK error code: Conflict in binding nonce recording.

	BS_SCSI_ERROR_DECOM_01 (see page 418)	IsoSDK error code: Cannot decompress using declared algorithm.
	BS_SCSI_ERROR_DISK_01 (see page 418)	IsoSDK error code: Incompatible medium installed.
	BS_SCSI_ERROR_DISK_02 (see page 418)	IsoSDK error code: Cannot read medium - unknown format.
	BS_SCSI_ERROR_DISK_03 (see page 418)	IsoSDK error code: Cannot read medium - incompatible format.
	BS_SCSI_ERROR_DISK_04 (see page 419)	IsoSDK error code: Cleaning cartridge installed.
	BS_SCSI_ERROR_DISK_05 (see page 419)	IsoSDK error code: Cannot write medium - unknown format.
	BS_SCSI_ERROR_DISK_06 (see page 419)	IsoSDK error code: Cannot write medium - incompatible format.
	BS_SCSI_ERROR_DISK_07 (see page 419)	IsoSDK error code: Cannot format medium - incompatible medium.
	BS_SCSI_ERROR_DISK_08 (see page 420)	IsoSDK error code: Cleaning failure.
	BS_SCSI_ERROR_DISK_09 (see page 420)	IsoSDK error code: Cannot write - application code mismatch.
	BS_SCSI_ERROR_DISK_10 (see page 420)	IsoSDK error code: Current session not fixated for append.
	BS_SCSI_ERROR_DISK_11 (see page 420)	IsoSDK error code: Medium not formatted.
	BS_SCSI_ERROR_DISK_12 (see page 421)	IsoSDK error code: Medium format corrupted.
	BS_SCSI_ERROR_DISK_13 (see page 421)	IsoSDK error code: Format command failed.

	BS_SCSI_ERROR_DISK_14 (see page 421)	IsoSDK error code: Medium not present.
	BS_SCSI_ERROR_DISK_15 (see page 421)	IsoSDK error code: Medium not present - tray closed.
	BS_SCSI_ERROR_DISK_16 (see page 422)	IsoSDK error code: Medium not present - tray open.
	BS_SCSI_ERROR_DISK_17 (see page 422)	IsoSDK error code: Medium not present - loadable.
	BS_SCSI_ERROR_DISK_18 (see page 422)	IsoSDK error code: Medium not present - medium auxiliary memory accessible.
	BS_SCSI_ERROR_DISK_19 (see page 422)	IsoSDK error code: Medium destination element full.
	BS_SCSI_ERROR_DISK_20 (see page 423)	IsoSDK error code: Medium source element empty.
	BS_SCSI_ERROR_DISK_21 (see page 423)	IsoSDK error code: End of medium reached.
	BS_SCSI_ERROR_DISK_22 (see page 423)	IsoSDK error code: Medium magazine not accessible.
	BS_SCSI_ERROR_DISK_23 (see page 423)	IsoSDK error code: Medium magazine removed.
	BS_SCSI_ERROR_DISK_24 (see page 424)	IsoSDK error code: Medium magazine inserted.
	BS_SCSI_ERROR_DISK_25 (see page 424)	IsoSDK error code: Medium magazine locked.
	BS_SCSI_ERROR_DISK_26 (see page 424)	IsoSDK error code: Medium magazine unlocked.
	BS_SCSI_ERROR_DISK_27 (see page 424)	IsoSDK error code: Medium loadable.
	BS_SCSI_ERROR_DISK_28 (see page 425)	IsoSDK error code: Medium auxiliary memory accessible.

	BS_SCSI_ERROR_DISK_29 (see page 425)	IsoSDK error code: The disc is not writable.
	BS_SCSI_ERROR_DISK_30 (see page 425)	IsoSDK error code: Wrong medium.
	BS_SCSI_ERROR_DISK_31 (see page 425)	IsoSDK error code: No medium.
	BS_SCSI_ERROR_DISK_32 (see page 426)	IsoSDK error code: The disc is not erasable.
	BS_SCSI_ERROR_DISK_33 (see page 426)	IsoSDK error code: Medium error.
	BS_SCSI_ERROR_DISK_34 (see page 426)	IsoSDK error code: Erase (see page 153) disc before writing.
	BS_SCSI_ERROR_DISK_35 (see page 426)	IsoSDK error code: The disc is not writable with current write method.
	BS_SCSI_ERROR_DISK_36 (see page 427)	IsoSDK error code: Cannot write medium - unsupported medium version.
	BS_SCSI_ERROR_DISK_37 (see page 427)	IsoSDK error code: Medium is not writeable.
	BS_SCSI_ERROR_DRIVE_01 (see page 427)	IsoSDK error code: Error reading UPC / EAN number.
	BS_SCSI_ERROR_DRIVE_02 (see page 427)	IsoSDK error code: Error reading ISRC number
	BS_SCSI_ERROR_EXT_01 (see page 428)	IsoSDK error code: Error detected by third party temporary initiator.
	BS_SCSI_ERROR_EXT_02 (see page 428)	IsoSDK error code: Third party device failure.
	BS_SCSI_ERROR_LOG_01 (see page 428)	IsoSDK error code: Error log overflow.
	BS_SCSI_ERROR_LOG_02 (see page 428)	IsoSDK error code: Log exception.
	BS_SCSI_ERROR_LOG_03 (see page 429)	IsoSDK error code: Log counter at maximum.

	BS_SCSI_ERROR_LOG_04 (see page 429)	IsoSDK error code: Log list codes exhausted.
	BS_SCSI_ERROR_MECH_01 (see page 429)	IsoSDK error code: Mechanical positioning error.
	BS_SCSI_ERROR_MECH_02 (see page 429)	IsoSDK error code: Positioning error detected by read of medium.
	BS_SCSI_ERROR_MECH_03 (see page 430)	IsoSDK error code: Mechanical positioning or changer error.
	BS_SCSI_ERROR_PARAM_01 (see page 430)	IsoSDK error code: Invalid field in parameter list.
	BS_SCSI_ERROR_PARAM_02 (see page 430)	IsoSDK error code: Parameter not supported.
	BS_SCSI_ERROR_PARAM_03 (see page 430)	IsoSDK error code: Parameter value invalid.
	BS_SCSI_ERROR_PARAM_04 (see page 431)	IsoSDK error code: Threshold parameters not supported.
	BS_SCSI_ERROR_PARAM_05 (see page 431)	IsoSDK error code: Parameters changed.
	BS_SCSI_ERROR_PARAM_06 (see page 431)	IsoSDK error code: Mode parameters changed.
	BS_SCSI_ERROR_PARAM_07 (see page 431)	IsoSDK error code: Log parameters changed.
	BS_SCSI_ERROR_PARAM_08 (see page 432)	IsoSDK error code: Rounded parameter.
	BS_SCSI_ERROR_PARAM_09 (see page 432)	IsoSDK error code: Saving parameters not supported.
	BS_SCSI_ERROR_PARAM_10 (see page 432)	IsoSDK error code: A bad parameter has been passed in.
	BS_SCSI_ERROR_READ_01 (see page 432)	IsoSDK error code: Unrecovered read error.

	BS_SCSI_ERROR_READ_02 (see page 433)	IsoSDK error code: Read retries exhausted.
	BS_SCSI_ERROR_RECOVER_01 (see page 433)	IsoSDK error code: Recovered data with no error correction applied.
	BS_SCSI_ERROR_RECOVER_02 (see page 433)	IsoSDK error code: Recovered data with retries.
	BS_SCSI_ERROR_RECOVER_03 (see page 433)	IsoSDK error code: Recovered data with positive head offset.
	BS_SCSI_ERROR_RECOVER_04 (see page 434)	IsoSDK error code: Recovered data with negative head offset.
	BS_SCSI_ERROR_RECOVER_05 (see page 434)	IsoSDK error code: Recovered data with retries and/or CIRC applied.
	BS_SCSI_ERROR_RECOVER_06 (see page 434)	IsoSDK error code: Recovered data using previous sector id.
	BS_SCSI_ERROR_RECOVER_07 (see page 434)	IsoSDK error code: Recovered data without ecc - recommend reassignment.
	BS_SCSI_ERROR_RECOVER_08 (see page 435)	IsoSDK error code: Recovered data without ecc - recommend rewrite.
	BS_SCSI_ERROR_RECOVER_09 (see page 435)	IsoSDK error code: Recovered data without ecc - data rewritten.
	BS_SCSI_ERROR_RECOVER_10 (see page 435)	IsoSDK error code: Recovered data with error correction applied.
	BS_SCSI_ERROR_RECOVER_11 (see page 435)	IsoSDK error code: Recovered data with error corr. & retries applied.
	BS_SCSI_ERROR_RECOVER_12 (see page 436)	IsoSDK error code: Recovered data - data auto-reallocated.

	BS_SCSI_ERROR_RECOVER_13 (see page 436)	IsoSDK error code: Recovered data with CIRC.
	BS_SCSI_ERROR_RECOVER_14 (see page 436)	IsoSDK error code: Recovered data with l-ec.
	BS_SCSI_ERROR_RECOVER_15 (see page 436)	IsoSDK error code: Recovered data - recommend reassignment.
	BS_SCSI_ERROR_RECOVER_16 (see page 437)	IsoSDK error code: Recovered data - recommend rewrite.
	BS_SCSI_ERROR_RECOVER_17 (see page 437)	IsoSDK error code: Recovered data with linking.
	BS_SCSI_ERROR_SEGM_01 (see page 437)	IsoSDK error code: Too many segment descriptors.
	BS_SCSI_ERROR_SEGM_02 (see page 437)	IsoSDK error code: Unsupported segment descriptor type code.
	BS_SCSI_ERROR_SEGM_03 (see page 438)	IsoSDK error code: Unexpected inexact segment.
	BS_SCSI_ERROR_SEGM_04 (see page 438)	IsoSDK error code: Copy segment granularity violation.
	BS_SCSI_ERROR_SESSION_01 (see page 438)	IsoSDK error code: Session fixation error.
	BS_SCSI_ERROR_SESSION_02 (see page 438)	IsoSDK error code: Session fixation error writing lead-in.
	BS_SCSI_ERROR_SESSION_03 (see page 439)	IsoSDK error code: Session fixation error writing lead-out.
	BS_SCSI_ERROR_SESSION_04 (see page 439)	IsoSDK error code: Session fixation error - incomplete track in session.
	BS_SCSI_ERROR_TARGET_01 (see page 439)	IsoSDK error code: Copy target device not reachable.

	BS_SCSI_ERROR_TARGET_02 (see page 439)	IsoSDK error code: Incorrect copy target device type.
	BS_SCSI_ERROR_TARGET_03 (see page 440)	IsoSDK error code: Copy target device data underrun.
	BS_SCSI_ERROR_TARGET_04 (see page 440)	IsoSDK error code: Session fixation error - incomplete track in session.
	BS_SCSI_ERROR_UNIT_01 (see page 440)	IsoSDK error code: No seek complete.
	BS_SCSI_ERROR_UNIT_02 (see page 440)	IsoSDK error code: Logical unit not ready, cause not reportable.
	BS_SCSI_ERROR_UNIT_03 (see page 441)	IsoSDK error code: Logical unit is in process of becoming ready.
	BS_SCSI_ERROR_UNIT_04 (see page 441)	IsoSDK error code: Logical unit not ready, initializing command required.
	BS_SCSI_ERROR_UNIT_05 (see page 441)	IsoSDK error code: Logical unit not ready, manual intervention required.
	BS_SCSI_ERROR_UNIT_06 (see page 441)	IsoSDK error code: Logical unit not ready, format in progress.
	BS_SCSI_ERROR_UNIT_07 (see page 442)	IsoSDK error code: Logical unit not ready, operation in progress.
	BS_SCSI_ERROR_UNIT_08 (see page 442)	IsoSDK error code: Logical unit not ready, long write in progress.
	BS_SCSI_ERROR_UNIT_09 (see page 442)	IsoSDK error code: Logical unit not ready, self-test in progress.

	BS_SCSI_ERROR_UNIT_10 (see page 442)	IsoSDK error code: Logical unit does not respond to selection.
	BS_SCSI_ERROR_UNIT_11 (see page 443)	IsoSDK error code: Logical unit communication failure.
	BS_SCSI_ERROR_UNIT_12 (see page 443)	IsoSDK error code: Logical unit communication time-out.
	BS_SCSI_ERROR_UNIT_13 (see page 443)	IsoSDK error code: Logical unit communication parity error.
	BS_SCSI_ERROR_UNIT_14 (see page 443)	IsoSDK error code: Logical unit communication CRC error (Ultra-DMA/32).
	BS_SCSI_ERROR_UNIT_16 (see page 444)	IsoSDK error code: Logical unit not supported.
	BS_SCSI_ERROR_UNIT_17 (see page 444)	IsoSDK error code: Logical unit has not self-configured yet.
	BS_SCSI_ERROR_UNIT_18 (see page 444)	IsoSDK error code: Logical unit failure.
	BS_SCSI_ERROR_UNIT_19 (see page 444)	IsoSDK error code: Timeout on logical unit.
	BS_SCSI_ERROR_UNIT_20 (see page 445)	IsoSDK error code: Logical unit failed self-test.
	BS_SCSI_ERROR_UNIT_21 (see page 445)	IsoSDK error code: Logical unit unable to update self-test log.
	BS_SCSI_ERROR_UNIT_22 (see page 445)	IsoSDK error code: Logical unit failed self-configuration.
	BS_SCSI_ERROR_UNIT_23 (see page 445)	IsoSDK error code: Logical unit failure prediction threshold exceeded.

	BS_SCSI_ERROR_VOL_01 (see page 446)	IsoSDK error code: Volume set created or modified.
	BS_SCSI_ERROR_VOL_02 (see page 446)	IsoSDK error code: Volume set deleted.
	BS_SCSI_ERROR_VOL_03 (see page 446)	IsoSDK error code: Volume set de-assigned.
	BS_SCSI_ERROR_VOL_04 (see page 446)	IsoSDK error code: Volume set reassigned.
	BS_SCSI_ERROR_WRITE_01 (see page 447)	IsoSDK error code: General write error.
	BS_SCSI_ERROR_WRITE_02 (see page 447)	IsoSDK error code: Write error - Recovery needed.
	BS_SCSI_ERROR_WRITE_03 (see page 447)	IsoSDK error code: Write error - Recovery failed.
	BS_SCSI_ERROR_WRITE_04 (see page 447)	IsoSDK error code: Write error - Loss of streaming.
	BS_SCSI_ERROR_WRITE_05 (see page 448)	IsoSDK error code: Write error - Padding blocks added.
	BS_SCSI_ERROR_WRITE_06 (see page 448)	IsoSDK error code: Write protected.
	BS_SCSI_ERROR_WRITE_07 (see page 448)	IsoSDK error code: Hardware write protected.
	BS_SCSI_ERROR_WRITE_08 (see page 448)	IsoSDK error code: Associated write protect.
	BS_SCSI_ERROR_WRITE_09 (see page 449)	IsoSDK error code: Persistent write protect.
	BS_SCSI_ERROR_WRITE_10 (see page 449)	IsoSDK error code: Permanent write protect.
	BS_SCSI_ERROR_WRITE_11 (see page 449)	IsoSDK error code: Conditional write protect.
	BS_SCSI_ERROR_WRITE_12 (see page 449)	IsoSDK error code: A write-error occurred.
	BS_SDK_COMPENC_BOTH (see page 450)	Medium Information: disc or image is compressed and encrypted.

	BS_SDK_COMPENC_COMPRESSED (see page 450)	Medium Information: disc or image is compressed.
	BS_SDK_COMPENC_ENCRYPTED (see page 450)	Medium Information: disc or image is encrypted.
	BS_SDK_CUE_ERROR_COMMAND_01 (see page 450)	IsoSDK error code: Invalid command.
	BS_SDK_CUE_ERROR_COMMAND_06 (see page 451)	IsoSDK error code: Command failed.
	BS_SDK_CUE_ERROR_FIELD (see page 451)	IsoSDK error code: Invalid field.
	BS_SDK_CUE_ERROR_FILE (see page 451)	IsoSDK error code: File not found.
	BS_SDK_CUE_ERROR_SENDING_CUE (see page 451)	IsoSDK error code: Send cue sheet failed.
	BS_SDK_CUE_ERROR_UEOl (see page 452)	IsoSDK error code: Unexpected end of line.
	BS_SDK_ERROR_ABORTED (see page 452)	IsoSDK error code: Operation aborted by user.
	BS_SDK_ERROR_ABORTEDCOMMAND (see page 452)	IsoSDK error code: Interface reported aborted command was thrown.
	BS_SDK_ERROR_BAD_CAPABILITY_NAME (see page 452)	Capabilities error: A non existing capability name was submitted.
	BS_SDK_ERROR_BAD_REQUEST (see page 453)	IsoSDK error code: Bad request.
	BS_SDK_ERROR_BIN_FILE_NOT_FOUND (see page 453)	IsoSDK error code: The bin file could not be found.
	BS_SDK_ERROR_BUFFERALIGNMENT (see page 453)	IsoSDK error code: Buffer alignment error.
	BS_SDK_ERROR_BUFFERTOOBIG (see page 453)	IsoSDK error code: The buffer is too big.
	BS_SDK_ERROR_BURN_IN_PROGRESS (see page 454)	IsoSDK error code: Not allowed while burning.

	BS_SDK_ERROR_CDTEXT_NOT_FOUND (see page 454)	IsoSDK error code: CDText is not available on the audio disc.
	BS_SDK_ERROR_CHECKCONDITION (see page 454)	IsoSDK error code: Check condition.
	BS_SDK_ERROR_COMPENC_GENERALERROR (see page 454)	IsoSDK error code: General error while read compress/encrypt identifier.
	BS_SDK_ERROR_COMPENC_READ10_IDENT (see page 455)	IsoSDK error code: No IsoSDK identifier found on disc or image.
	BS_SDK_ERROR_COMPRESSION_CONFLICT (see page 455)	IsoSDK error code: A compression conflict occurred.
	BS_SDK_ERROR_COMPRESSION_NOINDXTABLE (see page 455)	IsoSDK error code: No Indexes table found.
	BS_SDK_ERROR_COMPRESSION_NOSIGNATURE (see page 455)	IsoSDK error code: No IsoSDK compression signature found.
	BS_SDK_ERROR_COMPRESSION_READ10 (see page 456)	IsoSDK error code: Error while operate Read10 from image or disc. Maybe not data.
	BS_SDK_ERROR_COMPRESSION_READMEMF (see page 456)	IsoSDK error code: Error while read data from MemFile. Maybe empty.
	BS_SDK_ERROR_COMPRESSION_SEEKMEMFILE (see page 456)	IsoSDK error code: Error while seek inside MemFile. Maybe no data.
	BS_SDK_ERROR_COPYABORTED (see page 457)	IsoSDK error code: Copy aborted.
	BS_SDK_ERROR_CORRUPT_OR_INVALID_CUE_FILE (see page 457)	IsoSDK error code: The cue file is corrupt or completely invalid.
	BS_SDK_ERROR_CREATEFILE (see page 457)	IsoSDK error code: IsoSDK was not able to create the file.

	BS_SDK_ERROR_DATAOVERRUN ( <a href="#">see page 457</a> )	IsoSDK error code: A data-overrun occurred.
	BS_SDK_ERROR_DATAPROTECT ( <a href="#">see page 458</a> )	IsoSDK error code: Data protected.
	BS_SDK_ERROR_DATARECOVERERROR ( <a href="#">see page 458</a> )	IsoSDK error code: The IsoSDK was not able to recover the data.
	BS_SDK_ERROR_DEVICE_LOCKED ( <a href="#">see page 458</a> )	IsoSDK error code: The device is locked. Maybe by another software / process.
	BS_SDK_ERROR_EMPTY_PASSWORD ( <a href="#">see page 458</a> )	IsoSDK error code: Empty passwords are not allowed.
	BS_SDK_ERROR_ENCLIB_NOT_FOUND ( <a href="#">see page 459</a> )	IsoSDK error code: bass library was not found in directory.
	BS_SDK_ERROR_ENCRYPTION_CONFLICT ( <a href="#">see page 459</a> )	IsoSDK error code: Encryption conflict.
	BS_SDK_ERROR_ERASECHECK ( <a href="#">see page 459</a> )	IsoSDK error code: Erase ( <a href="#">see page 153</a> ) check.
	BS_SDK_ERROR_ERRINVALIDFILENAME ( <a href="#">see page 459</a> )	IsoSDK error code: Error importing session, file name length exceeds 120 chars.
	BS_SDK_ERROR_FILE_EXISTS ( <a href="#">see page 460</a> )	IsoSDK error code: The file already exists.
	BS_SDK_ERROR_FILE_OPEN ( <a href="#">see page 460</a> )	IsoSDK error code: File could not be opened.
	BS_SDK_ERROR_FILEINUSE ( <a href="#">see page 460</a> )	IsoSDK error code: File could not be opened.
	BS_SDK_ERROR_FILEMARK ( <a href="#">see page 460</a> )	IsoSDK error code: Filemark error.
	BS_SDK_ERROR_GENERAL ( <a href="#">see page 461</a> )	IsoSDK error code: A general error occurred.
	BS_SDK_ERROR_HARDWAREERROR ( <a href="#">see page 461</a> )	IsoSDK error code: Hardware error.

	BS_SDK_ERROR_ILLEGALLENGTH (see page 461)	IsoSDK error code: Illegal length of file name or path.
	BS_SDK_ERROR_IMPORTSESSION (see page 461)	IsoSDK error code: Error importing session.
	BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE (see page 462)	IsoSDK error code: Selected project type is incompatible with last recorded session.
	BS_SDK_ERROR_INCORRECTLENGTH (see page 462)	IsoSDK error code: Incorrect length of the given file name or path.
	BS_SDK_ERROR_INVALID_DEST_PATH (see page 462)	IsoSDK error code: An invalid destination path has been passed in.
	BS_SDK_ERROR_INVALID_DIR_INDEX (see page 462)	IsoSDK error code: Invalid directory index.
	BS_SDK_ERROR_INVALID_FILE_FORMAT (see page 463)	IsoSDK error code: The file has an invalid format.
	BS_SDK_ERROR_INVALID_FILE_NAME (see page 463)	IsoSDK error code: An invalid file name has been passed in.
	BS_SDK_ERROR_INVALID_HANDLE (see page 463)	IsoSDK error code: The submitted handle is not a capability handle or was not initialized with GetDeviceCapabilitiesHandle (see page 162)().
	BS_SDK_ERROR_INVALID_INDEX (see page 463)	IsoSDK error code: The SubIndex of the Audiotrack is wrong.
	BS_SDK_ERROR_INVALID_ISRC (see page 464)	IsoSDK error code: The ISRC code inside the track information is invalid.

	BS_SDK_ERROR_INVALID_MCN ( <a href="#">see page 464</a> )	IsoSDK error code: The MCN code inside the track information is invalid.
	BS_SDK_ERROR_INVALID_PATH ( <a href="#">see page 464</a> )	IsoSDK error code: An invalid path has been passed in.
	BS_SDK_ERROR_INVALID_SESSION_NUMBER ( <a href="#">see page 464</a> )	IsoSDK error code: Invalid session number has been passed in.
	BS_SDK_ERROR_INVALID_SRC_PATH ( <a href="#">see page 465</a> )	IsoSDK error code: An invalid source path has been passed in.
	BS_SDK_ERROR_INVALID_UDF_VERSION ( <a href="#">see page 465</a> )	IsoSDK error code: Invalid UDF version number.
	BS_SDK_ERROR_INVALIDSRB ( <a href="#">see page 465</a> )	IsoSDK error code: Invalid SRB.
	BS_SDK_ERROR_ISOIMAGENOTFOUND ( <a href="#">see page 465</a> )	IsoSDK error code: Error open ISO image file. ISO image not found.
	BS_SDK_ERROR_MAXDIRS ( <a href="#">see page 466</a> )	IsoSDK error code: Not more than %d directories are allowed.
	BS_SDK_ERROR_MAXFILES ( <a href="#">see page 466</a> )	IsoSDK error code: Not more than %d files are allowed.
	BS_SDK_ERROR_MESSAGEREJECT ( <a href="#">see page 466</a> )	IsoSDK error code: Message has been rejected.
	BS_SDK_ERROR_MISCOMPARE ( <a href="#">see page 466</a> )	IsoSDK error code: Miscompare. Compared names are different.
	BS_SDK_ERROR_MORE_SPACE_NEEDED ( <a href="#">see page 467</a> )	IsoSDK error code: The supplied buffer is too small.
	BS_SDK_ERROR_MP3LIB_NOT_FOUND ( <a href="#">see page 467</a> )	IsoSDK error code: mp3lib (FoxPlayer) not found.

	BS_SDK_ERROR_NETTAGS_CONNECT (see page 467)	IsoSDK error code: Error while connect to the CDDB server.
	BS_SDK_ERROR_NETTAGS_DISK (see page 467)	IsoSDK error code: Compact disc error during forming request to internet CDDB tags.
	BS_SDK_ERROR_NETTAGS_INTERNAL (see page 468)	IsoSDK error code: A internal error occurred. Please contact our support.
	BS_SDK_ERROR_NETTAGS_NOMATCH (see page 468)	IsoSDK error code: No match while searching CDDB database.
	BS_SDK_ERROR_NETTAGS_SERVER (see page 468)	IsoSDK error code: A server error occur. With FreeDB try another mirror.
	BS_SDK_ERROR_NEXTADDRESS (see page 468)	IsoSDK error code: Error getting next writable address.
	BS_SDK_ERROR_NO (see page 469)	IsoSDK error code: No error occurred.
	BS_SDK_ERROR_NOT_ALLOWED (see page 469)	IsoSDK error code: Operation / command not allowed.
	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER (see page 469)	IsoSDK error code: Not allowed for the current burner.
<span style="background-color: #e0f2e0;">new</span>	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE (see page 469)	IsoSDK error code: If you set a forbidden option or function according to the used project.
	BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION (see page 470)	IsoSDK error code: Invalid option for this UDF version.
	BS_SDK_ERROR_NOT_IMPLEMENTED (see page 470)	IsoSDK error code: The feature ( command is currently not implemented.
	BS_SDK_ERROR_NOTREADY (see page 470)	IsoSDK error code: Device not ready.

	BS_SDK_ERROR_NOTSUPPORTED (see page 471)	IsoSDK error code: Operation / command not supported.
	BS_SDK_ERROR_PARITYERR (see page 471)	IsoSDK error code: Partition version error.
	BS_SDK_ERROR_PATH_EXISTS (see page 471)	IsoSDK error code: The path already exists.
	BS_SDK_ERROR_PLUGIN (see page 471)	IsoSDK error code: Error while loading external plugin.
	BS_SDK_ERROR_RESERVED (see page 472)	IsoSDK error code: Reserved.
	BS_SDK_ERROR_SELECTIONTIMEOUT (see page 472)	IsoSDK error code: Selection timed out.
	BS_SDK_ERROR_SRBTIMEOUT (see page 472)	IsoSDK error code: A SRB timeout occurred.
	BS_SDK_ERROR_TIMEOUT (see page 472)	IsoSDK error code: A timeout occurred.
	BS_SDK_ERROR_TOOMUCH_DATA (see page 473)	IsoSDK error code: Too much data for this mode. The size of the collection will exceed the disc size.
	BS_SDK_ERROR_TOOMUCH_INDEXES (see page 473)	IsoSDK error code: To much SubIndex values found. Max is 99.
	BS_SDK_ERROR_UNEXPECTEDBUSFREE (see page 473)	IsoSDK error code: Unexpected bus free.
	BS_SDK_ERROR_UNITATTENTION (see page 473)	IsoSDK error code: Unit attention.
	BS_SDK_ERROR_UNKNOWN (see page 474)	IsoSDK error code: An unknown error occurred.
	BS_SDK_ERROR_UNKNOWN_TEXTID (see page 474)	IsoSDK error code: The text ID that has been passed in is unknown.
	BS_SDK_ERROR_UNSUPPORTED_MEDIUM (see page 474)	IsoSDK error code: A medium was inserted that is not supported by IsoSDK.
	BS_SDK_ERROR_VOLUMEOVERFLOW (see page 474)	IsoSDK error code: Volume overflow.

	BS_SDK_INT_ERROR_1 (see page 475)	IsoSDK error code: Internal error 1.
	BS_SDK_INT_ERROR_2 (see page 475)	IsoSDK error code: Internal error 2.
	BS_SDK_INT_ERROR_3 (see page 475)	IsoSDK error code: Internal error 3.
	BS_SDK_INT_ERROR_4 (see page 475)	IsoSDK error code: Internal error 4.
	BS_SDK_INT_ERROR_5 (see page 476)	IsoSDK error code: Internal error 5.
	BS_SDK_INT_ERROR_FORMAT (see page 476)	IsoSDK error code: Error formatting re-writeable medium.
	BS_SDK_KEY_INVALID (see page 476)	IsoSDK message: An invalid license key was submitted.
	BS_SDK_KEY_VALID (see page 476)	IsoSDK message: The license key is valid.
	BS_SDK_MESSAGE_01 (see page 477)	IsoSDK Message: Text unknown
	BS_SDK_MESSAGE_02 (see page 477)	IsoSDK message: Waiting for user interaction
	BS_SDK_MESSAGE_03 (see page 477)	IsoSDK message: Cannot close dialog
	BS_SDK_MESSAGE_04 (see page 478)	IsoSDK message: Please stop the process first.
	BS_SDK_MESSAGE_05 (see page 478)	IsoSDK message: Erasing CD/DVD ...
	BS_SDK_MESSAGE_06 (see page 478)	IsoSDK message: Preparing Data ...
	BS_SDK_MESSAGE_07 (see page 479)	IsoSDK message: Finalizing CD/DVD ...
	BS_SDK_MESSAGE_08 (see page 479)	IsoSDK message: Burning CD/DVD ...

	BS_SDK_MESSAGE_10 (see page 479)	IsoSDK message: Aborting Process ...
	BS_SDK_MESSAGE_11 (see page 480)	IsoSDK message: Process successfully completed.
	BS_SDK_MESSAGE_12 (see page 480)	IsoSDK message: max.
	BS_SDK_MESSAGE_13 (see page 480)	IsoSDK message: Please insert the next CD/DVD.
	BS_SDK_MESSAGE_14 (see page 481)	IsoSDK message: Starting the verify process ...
	BS_SDK_MESSAGE_15 (see page 481)	IsoSDK message: Process completed with %d error(s).
	BS_SDK_MESSAGE_16 (see page 481)	IsoSDK message: Verifying file.
<span style="color: green;">new</span>	BS_SDK_MESSAGE_COMPENC_BOTH (see page 482)	The IsoSDK will throw this message if the signature for compression and encryption was found.
<span style="color: green;">new</span>	BS_SDK_MESSAGE_COMPENC_COMPRESSED (see page 482)	The IsoSDK will throw this message if the signature for compression was found.
<span style="color: green;">new</span>	BS_SDK_MESSAGE_COMPENC_ENCRYPTED (see page 482)	The IsoSDK will throw this message if the signature for encryption was found.
	BS_SDK_MESSAGE_ERASESTART (see page 482)	IsoSDK message: Starting the erase process.
	BS_SDK_MESSAGE_EXTR_FILE (see page 483)	IsoSDK message: Extracting file.
	BS_SDK_MESSAGE_FORMAT (see page 483)	IsoSDK message: Formatting.

	BS_SDK_MESSAGE_FORMAT_DONE (see page 483)	IsoSDK message: Formatting is done. Thrown after BS_SDK_MESSAGE_FORMAT (see page 483).
	BS_SDK_MESSAGE_IMAGECREATESTART (see page 484)	IsoSDK message: Image creation started.
	BS_SDK_MESSAGE_IMPORT (see page 484)	IsoSDK message: Importing session.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY (see page 484)	IsoSDK message: Database entry can't be selected.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_MATCHING_PARAM (see page 485)	IsoSDK message:
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CREATING_PROVIDER (see page 485)	IsoSDK message: Error creating tag provider: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES (see page 486)	IsoSDK message: Error getting matching entries: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY (see page 486)	IsoSDK message: Error getting selected entry: %s
	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY (see page 487)	IsoSDK message: Fetching selected entry, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES (see page 487)	IsoSDK message: Getting matching entries, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS (see page 488)	IsoSDK message: Getting matching parameters, please wait...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE (see page 488)	IsoSDK message: Initialization complete. Processing...
	BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES (see page 489)	IsoSDK message: Multiple matches.

	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY (see page 489)	IsoSDK message: Please, select an entry from the list below.
	BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH (see page 490)	IsoSDK message: Single match.
	BS_SDK_MESSAGE_SIMULATE (see page 490)	IsoSDK message: Test write.
	BS_SDK_MESSAGE_WAIT (see page 490)	IsoSDK message: Writing lead-out (may take 15-20 minutes).
	BS_SDK_MESSAGE_WRITESTART (see page 491)	IsoSDK message: Starting write process.
	BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE (see page 491)	IsoSDK error code: The burned file is not readable.
	BS_SDK_VERIFY_ERROR_FILESDIFFERENT (see page 491)	IsoSDK error code: The burned file differs from the source file.
	BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE (see page 491)	IsoSDK error code: The source file is not readable.
	BS_SEPARATE_CHANNEL_MUTE (see page 492)	Drive feature: The Logical Unit shall support independently muting each audio channel via the CD Audio Control Page.
	BS_SEPARATE_VOLUME_LEVELS (see page 492)	Drive feature: The Logical Unit shall support separately controllable audio levels for each supported channel via the CD Audio Control Page.
	BS_SMART (see page 492)	Drive feature: Device is able to perform Self-Monitoring Analysis and Reporting Technology.

	BS_STREAMING (see page 492)	Drive feature: Device is able to perform reading and writing within specified performance ranges.
<span style="background-color: #90EE90; color: black; padding: 2px;">new</span>	BS_STRING_MANIPULATION_DETECTED (see page 493)	The IsoSDK will check the string length inside the function body of dedicated classes. If a wrong string length was found the IsoSDK will assume a manipulation and will stop to avoid attacks.
	BS_TCH_CDTEXT (see page 493)	Audio grabber tag information: Receive tags only from CD-Text.
	BS_TCH_CDTEXT_INTERNETDB (see page 493)	Audio grabber tag information: Receive tags first from CD-Text then from FreeDB / CDDB.
	BS_TCH_INTERNETDB (see page 493)	Audio grabber tag information: Receive tags only from FreeDB / CDDB
	BS_TCH_INTERNETDB_CDTEXT (see page 494)	Audio grabber tag information: Receive tags First from FreeDB/CDDB then from CD-Text.
	BS_TCH_NONE (see page 494)	Audio grabber tag information: Receive no tags.
	BS_TCI_ARTIST (see page 494)	Audio disc CDDB tag information: Artist information.
	BS_TCI_CATEGORY (see page 494)	Audio disc CDDB tag information: Category information.
	BS_TCI_DISK_LENGTH (see page 495)	Audio disc CDDB tag information: disc length information

	BS_TCI_DISKID (see page 495)	Audio disc CDDB tag information: Get the Disc id.
	BS_TCI_EXTENDED_INFO (see page 495)	Audio disc CDDB tag information: Extended information.
	BS_TCI_FRAME_OFFSET (see page 495)	Audio disc CDDB tag information: Frame offset information.
	BS_TCI_GENRE (see page 496)	Audio disc CDDB tag information: Genre information.
	BS_TCI_REVISION (see page 496)	Audio disc CDDB tag information: Revision information.
	BS_TCI_SUBMITTED_VIA (see page 496)	Audio disc CDDB tag information: Receive information about the identifier that was used to submit data.
	BS_TCI_TITLE (see page 496)	Audio disc CDDB tag information: Receive track or disc title.
	BS_TCI_YEAR (see page 497)	Audio disc CDDB tag information: Receive production year.
	BS_TF_AUDIO (see page 497)	Track type: Audio track.
	BS_TF_DATA_MODE1 (see page 497)	Track type: Mode 1 track (always for DVD/BD).
	BS_TF_DATA_MODE2 (see page 497)	Track type: Mode 2 track.
	BS_TM_DMA (see page 498)	IsoSDK transfer mode: Direct Memory Access
	BS_TM_NOT_APPLICABLE (see page 498)	IsoSDK transfer mode: Not possible to evaluate or set.

	BS_TM_PIO (see page 498)	IsoSDK transfer mode: Programmed Input/Output
	BS_TM_UNKNOWN (see page 498)	IsoSDK transfer mode: Unknown
	BS_TRUE (see page 499)	IsoSDK BOOL value: TRUE
	BS_UDF_PARTITION_PHYSICAL (see page 499)	UDF partition type: Physical or Type 1 This is the simplest partition. A type 1 partition has a start address S and size N. A logical block number A in the partition can be converted to the media physical address (in UDF's term, the logical sector address) S+A. In certain optical media, the start and size of the partition must be aligned to the packet size (such as 32KB). These special requirements are defined in the appendixes of the UDF standard. Free space of the partition is managed by the Unallocated Space Bitmap Descriptor. It contains one bit for each... more (see page 499)

	BS_UDF_PARTITION_SPARABLE (see page 499)	UDF partition type: Sparable. Sparable partitions are used on overwrite media that will fail after a certain number of overwrites (several thousands), such as CD-RW. In a file system, the places that are overwritten frequently are often important metadata area, e.g., bitmaps. Sparable partition allows the failed area to be remapped to other good part on the media so the failed area appears good to the upper level. A sparable partition is similar to a type 1 partition in the sense that it has a start address and size. Moreover, it defines 2 to 4 sparing tables which points to... more (see page 499)
--	--	--

	BS_UDF_PARTITION_VIRTUAL (see page 500)	UDF partition type: Virtual. Virtual partition is used on write-once media. Only three types of metadata are stored in the virtual partition: File Set Descriptor, File Entry (including Extended File Entry), and Allocation Extent Descriptor. If the file data is embedded in the file entry, these file data are also stored in the virtual partition. Virtual partition makes the write-once media appear as an overwrite media. Virtual partition layers on top of the type 1 partition. A Virtual Allocation Table (VAT) is used to map logical addresses of the virtual partition to logical addresses in the underlying type 1 partition.... more (see page 500)
	BS_UDF_VERSION_102 (see page 500)	UDF version: 1.02 - VideoDVD
	BS_UDF_VERSION_150 (see page 500)	UDF version: 1.50
	BS_UDF_VERSION_200 (see page 501)	UDF version: 2.00
	BS_UDF_VERSION_201 (see page 501)	UDF version: 2.01
	BS_UDF_VERSION_250 (see page 501)	UDF version: 2.50
	BS_UDF_VERSION_260 (see page 501)	UDF version: 2.60
	BS_UDFOPT_ALL (see page 502)	UDF options: Set all options.
	BS_UDFOPT_AVCHD_DISC (see page 502)	UDF options: Set AVCHD flag.
	BS_UDFOPT_FILE_STREAMS (see page 502)	UDF options: Set write file streams.

	BS_UDFOPT_IMPLEMENTATION_ID (see page 502)	UDF options: Write implementation ID.
	BS_UDFOPT_PARTITION_TYPE (see page 503)	UDF options: Write partition type.
	BS_UDFOPT_VERSION (see page 503)	UDF options: Write UDF version.
	BS_UNDERRUN_PROTECTION (see page 503)	Drive feature: Device is able to perform buffer protection functions like <b>Burn-Proof</b> .
	BS_UNKNOWN (see page 503)	A disc identifier for disc type: Unknown
	BS_UPC_READ (see page 504)	Drive feature: Device is able to read the UPC info of a disc.
	BS_VCD_INFINITE (see page 504)	This is macro BS_VCD_INFINITE.
	BS_VCD_KEY_0 (see page 504)	This is macro BS_VCD_KEY_0 .
	BS_VCD_KEY_DEFAULT (see page 504)	This is macro BS_VCD_KEY_DEFAULT.
	BS_VCD_KEY_NEXT (see page 505)	This is macro BS_VCD_KEY_NEXT.
	BS_VCD_KEY_PREVIOUS (see page 505)	This is macro BS_VCD_KEY_PREVIOUS.
	BS_VCD_KEY_RETURN (see page 505)	This is macro BS_VCD_KEY_RETURN.
	BS_VCD_SEGMENT_ITEM_0 (see page 505)	This is macro BS_VCD_SEGMENT_ITEM_0.
	BS_VCD_TRACK_ITEM_0 (see page 506)	This is macro BS_VCD_TRACK_ITEM_0.
	BS_WHOLE_PATH (see page 506)	The complete path of the file is created as a subfolder in the chDestinationPath of the project.
	BS_WM_DAO (see page 506)	Write method: DAO (Disc-At-Once)

	BS_WM.DAO96 (see page 506)	Write method: DAO96 (Disc-At-Once + 96)
	BS_WM.TAO (see page 507)	Write method: TAO (Track-At-Once)
	BS_WRITE_BLURAY_R (see page 507)	Drive feature: Device is able to write Blu-Ray Recordable.
	BS_WRITE_BLURAY_R_XL (see page 507)	Drive feature: Device is able to write BD-R XL.
	BS_WRITE_BLURAY_RE (see page 507)	Drive feature: Device is able to write Blu-Ray ReWritable.
	BS_WRITE_BLURAY_RE_XL (see page 508)	Drive feature: Device is able to write BD-RE XL (Re-Writable).
	BS_WRITE_CDR (see page 508)	Drive feature: Device is able to write CD-Recordable.
	BS_WRITE_CDRW (see page 508)	Drive feature: Device is able to write CD-ReWritable.
	BS_WRITE_CDRW_CAV (see page 508)	Drive feature: Device is able to write CD-RW media that is designed for CAV recording.
	BS_WRITE_DVD_DL (see page 509)	Drive feature: Device is able to write DVD-R Double Layer.
	BS_WRITE_DVD_MRDL (see page 509)	Drive feature: Device is able to write DVD-R Double Layer.
	BS_WRITE_DVD_RDL_PLUS (see page 509)	Drive feature: Device is able to write DVD+R Double Layer.
	BS_WRITE_DVD_RWDL_PLUS (see page 509)	Drive feature: Device is able to write DVD+ReWritable Double Layer.

	BS_WRITE_DVDR (see page 510)	Drive feature: Device is able to write DVD-R.
	BS_WRITE_DVDR_PLUS (see page 510)	Drive feature: Device is able to write DVD+R.
	BS_WRITE_DVDRAM (see page 510)	Drive feature: Device is able to write DVD-RAM.
	BS_WRITE_DVDRW (see page 510)	Drive feature: Device is able to write DVD-ReWritable.
	BS_WRITE_DVDRW_PLUS (see page 511)	Drive feature: Device is able to write DVD+ReWritable.
	BS_WRITE_HDDVD_R (see page 511)	Drive feature: Device is able to write HDDVD Reordable.
	BS_WRITE_HDDVD_RW (see page 511)	Drive feature: Device is able to write HDDVD ReWritable.
	BS_WRITE_MOUNT_RAINER (see page 511)	Drive feature: Device is able to write Mount Rainer.
	BS_WRITE_TEST (see page 512)	Drive feature: Device is able to simulate burning.
	COMPRESS_BLOCKSIZE (see page 512)	This is macro COMPRESS_BLOCKSIZE.
	ENCRYPT_BLOCKSIZE (see page 512)	This is macro ENCRYPT_BLOCKSIZE.
	INSERT_STRUCTURE_PADDING (see page 513)	This is macro INSERT_STRUCTURE_PADDING.
	int2bool (see page 513)	This is macro int2bool.
	int2BS_BOOL (see page 513)	This is macro int2BS_BOOL.
	PATHSEPSTRING (see page 513)	This is macro PATHSEPSTRING.
	STRUCTURE_PADDING_BIG (see page 515)	This is macro STRUCTURE_PADDING_BIG.
	STRUCTURE_PADDING_HUGE (see page 515)	This is macro STRUCTURE_PADDING_HUGE.

	STRUCTURE_PADDING_NORMAL (see page 515)	This is macro STRUCTURE_P ADDING_NORM AL.
	STRUCTURE_PADDING_SMALL (see page 516)	This is macro STRUCTURE_P ADDING_SMALL .
	STRUCTURE_PADDING_STRING_DATA (see page 516)	This is macro STRUCTURE_P ADDING_STRIN G_DATA.

## Structures

	Name	Description
	_FoxEntries (see page 231)	This is record _FoxEntries.
	_FoxIndecies (see page 231)	This is record _FoxIndecies.
	SAudioGrabbingParams (see page 2)	IsoSDK sub class that contains information for audio grabbing
	SBootInfoEx (see page 6)	IsoSDK sub class is used to get and set the parameters for boot disc.
	SBootVolumeInfo (see page 10)	IsoSDK sub class that contains the information about the boot volume information.
	SCompressEncryptInfo (see page 11)	IsoSDK sub class that is used to set and get the parameters for a compressed and/or encrypted data disc.
	SCreateImageParams (see page 14)	IsoSDK sub class that contains information for the ImageCreate operation.
	SDirToAdd (see page 16)	IsoSDK sub class that contains the information about the directory you want to add to the project.
	SDirToCreate (see page 20)	IsoSDK sub class that contains the information about the directory you want to create inside the project.
	SDirToRemove (see page 21)	IsoSDK sub class that contains the information about the directory to be removed.
	SDirRename (see page 23)	IsoSDK sub class that contains the information about the directory to be renamed.
	SDiskCopyOptions (see page 24)	IsoSDK sub class that contains information for the DiskCopy operation.
	SDVDVideoOptions (see page 28)	IsoSDK sub class that contains additional information for VideoDVD projects.
	SExtendedDeviceInformation (see page 29)	IsoSDK sub class that contains the extended information for the device.
	SExtent (see page 231)	A structure that contains the information about the Extent information of a file in allocation table.
	SFileAllocationTable (see page 232)	A structure that contains the information about the file allocation table.
	SFileAudioProperty (see page 34)	IsoSDK sub class that contains information about CD-Text and track properties.
	SFileDateTime (see page 38)	IsoSDK sub class that contains the information about the time & date of the specific file.
	SFileEntry (see page 40)	IsoSDK sub class that contains the information about the file to be used.
	SFileTypeEx (see page 44)	<p>This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.</p> <p>You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes (see page 211) method.</p> <p>This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.</p>

	SFileToAdd ( <a href="#">see page 47</a> )	IsoSDK sub class that contains the information about the file you want to add to the project.
	SFileToAddEx ( <a href="#">see page 50</a> )	IsoSDK sub class that contains the information about the file segment you want to add to the project.
	SFileToRemove ( <a href="#">see page 53</a> )	IsoSDK sub class that contains the information about the file to be removed.
	SFileToRename ( <a href="#">see page 54</a> )	IsoSDK sub class contains the information about the file to be renamed.
	SISOInfoEx ( <a href="#">see page 56</a> )	This class is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.  You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes ( <a href="#">see page 211</a> ) method.  This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.
	SISOVolumeInfo ( <a href="#">see page 64</a> )	IsoSDK sub class that contains the information about the ISO information.
	SMediumInfo ( <a href="#">see page 67</a> )	IsoSDK sub class that contains the information about the medium.
	SOPTIONS ( <a href="#">see page 75</a> )	IsoSDK sub class that is used to get and set the parameters for the current project.
	SRawTrackToAdd ( <a href="#">see page 82</a> )	IsoSDK sub class that contains the information of a RAW image.
	SReadErrorCorrectionParams ( <a href="#">see page 86</a> )	IsoSDK sub class that contains the information of the ReadErrorCorrection for DiskCopy and CreateImage ( <a href="#">see page 127</a> ).
	SSessionInfo ( <a href="#">see page 88</a> )	IsoSDK sub class that contains the information about the selected session.
	SSpeed ( <a href="#">see page 90</a> )	IsoSDK sub class that contains information about the possible burning speeds.
	STags ( <a href="#">see page 92</a> )	IsoSDK sub class that
	STrackInfo ( <a href="#">see page 94</a> )	IsoSDK sub class that contains the information about the selected track.
	SUDFOptions ( <a href="#">see page 97</a> )	IsoSDK sub class that is used to get and set the parameters for UDF disc.
	SUDFOptionsEx ( <a href="#">see page 232</a> )	This structure is used to get and set the extended parameters for UDF disc. Added StructSize fields to the structure. This field must be filled-in with structure size by the caller. For example:
	SUDFVolumeInfo ( <a href="#">see page 100</a> )	IsoSDK sub class that contains the information about the UDF information.
	SVideoFormat ( <a href="#">see page 105</a> )	IsoSDK sub class that contains information of a mpeg video file.
	FoxEntries ( <a href="#">see page 233</a> )	This is type FoxEntries.
	FoxIndecies ( <a href="#">see page 233</a> )	This is type FoxIndecies.
	PFoxEntries ( <a href="#">see page 234</a> )	This is type PFoxEntries.
	PFoxIndecies ( <a href="#">see page 234</a> )	This is type PFoxIndecies.

## Types

Name	Description
AddFileEvent ( <a href="#">see page 235</a> )	This event will be triggered when a file was successfully added to the project.
AudioDecodeDoneEvent ( <a href="#">see page 236</a> )	This event will be triggered after a audio file was decoded.
AudioDecoderEvent ( <a href="#">see page 236</a> )	This event will be triggered during the decoding of a audio file. It shows the current progress of the process.
BurnDoneEvent ( <a href="#">see page 237</a> )	This event will be triggered when the burning process was completed.
BurnFileEvent ( <a href="#">see page 237</a> )	This event will be triggered for each file in the current burning process.
CompareFilesForArrangementEvent ( <a href="#">see page 237</a> )	This event will be triggered after a new dir was create in the current project.
CreateDirEvent ( <a href="#">see page 238</a> )	This event will be triggered after a new dir was create in the current project.

EraseDoneEvent ( <a href="#">see page 238</a> )	This event will be triggered when the deletion process was completed.
FinalizeEvent ( <a href="#">see page 239</a> )	This event will be triggered during the burning process when the finalize process starts. After this event is triggered the Event ProcessEvent ( <a href="#">see page 242</a> ) will report the progress of the finalize process.
GetTextEvent ( <a href="#">see page 239</a> )	This event will be triggered when you call the GetText ( <a href="#">see page 181</a> ()) function.
HDIR ( <a href="#">see page 239</a> )	HDIR is a virtual directory's handle. It is needed to read contents of disc session and IsoSDK's project. HDIR handles are obtained using OpenDirectory ( <a href="#">see page 192</a> ()) function. After finished using HDIR handle it should be closed by CloseDirectory ( <a href="#">see page 124</a> ()) function.
HSESSION ( <a href="#">see page 240</a> )	HSESSION is a virtual session handle. It is needed to list disc session and IsoSDK's projects contents, to read file system information, to import, read and verify files from disc. It is returned by OpenDiskSession ( <a href="#">see page 193</a> ()) function. After finished using HSESSION handle it should be closed by CloseDiskSession ( <a href="#">see page 125</a> ()).
InfoTextEvent ( <a href="#">see page 240</a> )	This event is triggered when the IsoSDK was loaded and an action was performed. This event is useable for logging.
int16 ( <a href="#">see page 240</a> )	This is type int16.
int32 ( <a href="#">see page 241</a> )	This is type int32.
int64 ( <a href="#">see page 241</a> )	This is type int64.
int8 ( <a href="#">see page 241</a> )	This is type int8.
JobDoneEvent ( <a href="#">see page 241</a> )	This event will be triggered when all operations with the device are finished (after burn or erase process) and the IsoSDK is available for further use.
ProcessEvent ( <a href="#">see page 242</a> )	This event will be triggered during a deletion or burn, erase and finalize process. It shows the current progress of the process.
RAWDataEvent ( <a href="#">see page 242</a> )	This event will be triggered if the IsoSDK needs more data for burning the RAW Image file. This event will only occur if no file was added to a BS_BRUNER_RAW project.
RemoveFileEvent ( <a href="#">see page 243</a> )	This event will be triggered when a file was removed from the project.
StartVerifyEvent ( <a href="#">see page 243</a> )	This event will be triggered when the verify process will start.
uint16 ( <a href="#">see page 243</a> )	This is type uint16.
uint32 ( <a href="#">see page 244</a> )	This is type uint32.
uint8 ( <a href="#">see page 244</a> )	This is type uint8.
VerifyDoneEvent ( <a href="#">see page 244</a> )	This event will be triggered when the verify process is finished.
VerifyErrorEvent ( <a href="#">see page 245</a> )	This event is triggered if an error occurred during the current verify process.
VerifyFileEvent ( <a href="#">see page 245</a> )	This event will be triggered every time a new file will be verified.
VerifySectorEvent ( <a href="#">see page 245</a> )	This event will be triggered every time a new file will be verified.
VideoScanDoneEvent ( <a href="#">see page 246</a> )	This event will be triggered when the video scanning process is completed.
VideoScannerEvent ( <a href="#">see page 246</a> )	This event will be triggered during a running scanning process. It shows the current progress of the process.

## 1.6.3 IsoSDKExport.h

This is file IsoSDKExport.h.

### Functions

	Name	Description
Abort ( <a href="#">see page 118</a> )		This function interrupts the current burning process.
AddBurnDevice ( <a href="#">see page 118</a> )		This function will add a device to the multi burn device list. This is important for multi-device burning.

	AddDir ( <a href="#">see page 118</a> )	This function adds a directory and its contents to your project.
	AddFile ( <a href="#">see page 119</a> )	This function adds a single file to your project.
	AddFileEx ( <a href="#">see page 120</a> )	This function allow you to add defined segments of a file to the project.
	AnalyseDeviceCapability ( <a href="#">see page 120</a> )	With this function you can analyze all possible capabilities of the current capability handle. You can analyze as many capabilities and as many times as needed.
	AudioFileStop ( <a href="#">see page 121</a> )	This function stops the internal audio player.
	Burn ( <a href="#">see page 121</a> )	This function writes the prepared project to the inserted disc.
	BurnDialog ( <a href="#">see page 121</a> )	This function starts the burning process with the embedded dialog.
	BurnISO ( <a href="#">see page 122</a> )	This function writes a defined ISO file to a disc.
	CheckLicenseKey ( <a href="#">see page 123</a> )	The function CheckLicenseKey validates the given key.
	CheckSignature ( <a href="#">see page 123</a> )	Function to check if the disc or image has a signature for encryption or compression.
	ClearAll ( <a href="#">see page 123</a> )	This function removes all files and folders from the current project.
	CloseCDTextHandle ( <a href="#">see page 124</a> )	This function will close a created handle to the CD-Text information of the audio disc.
	CloseDevice ( <a href="#">see page 124</a> )	This function closes the current device (tray).
	CloseDirectory ( <a href="#">see page 124</a> )	This function closes an open directory handle.
	CloseDiskSession ( <a href="#">see page 125</a> )	This function closes an open session handle.
	CloseNetworkTagsHandle ( <a href="#">see page 125</a> )	This function will close a CDDA tag handle that was initialized.
	CloseSession ( <a href="#">see page 126</a> )	This function tries to close the last open session on the current disc.
	ConvertSpeedFromKBPerSec ( <a href="#">see page 126</a> )	This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0.
	CopyDisk ( <a href="#">see page 127</a> )	This function will copy a disc directly to a selected burner.
	CreateDir ( <a href="#">see page 127</a> )	This function creates a new empty directory in the current project.
	CreateImage ( <a href="#">see page 127</a> )	This function will copy a disc to a image file.
	CreateProject ( <a href="#">see page 128</a> )	This function creates a new project. The current project will be deleted.
<b>new</b>	DABurn ( <a href="#">see page 129</a> )	This function writes the prepared project to the inserted disc. <b>Direct access version.</b>
<b>new</b>	DABurnISOImage ( <a href="#">see page 129</a> )	This function writes a defined ISO file to a disc. <b>Direct access version.</b>
<b>new</b>	DACheckSignature ( <a href="#">see page 130</a> )	Function to check if the disc or image has a signature for encryption or compression. <b>Direct access version.</b>
<b>new</b>	DACloseDevice ( <a href="#">see page 130</a> )	This function closes the current device (tray). <b>Direct access version.</b>
<b>new</b>	DACloseSession ( <a href="#">see page 130</a> )	This function tries to close the last open session on the current disc. <b>Direct access version.</b>
<b>new</b>	DACConvertSpeedFromKBPerSec ( <a href="#">see page 131</a> )	This function will convert/calculate a given speed in kb/s, like 7200 kb/s, to a readable speed information, like 48.0. <b>Direct access version.</b>
<b>new</b>	DACopyDisk ( <a href="#">see page 131</a> )	This function will copy a disc directly to a selected burner. <b>Direct access version.</b>
<b>new</b>	DACreateImage ( <a href="#">see page 132</a> )	This function will copy a disc to a image file. <b>Direct access version.</b>

 <b>DAEjectDevice</b> ( <a href="#">see page 132</a> )	This function opens the current device (tray). <b>Direct access version.</b>
 <b>DAErase</b> ( <a href="#">see page 133</a> )	This function erases the disc in the current device. <b>Direct access version.</b>
 <b>DAGetBurnSpeed</b> ( <a href="#">see page 133</a> )	This function returns the writing speed of the selected device. <b>Direct access version.</b>
 <b>DAGetDeviceCapabilities</b> ( <a href="#">see page 134</a> )	If you need to know the device capabilities use this function. <b>Direct access version.</b> <b>DEPRECATED! IS RELPACED BY:</b> AnalyseDeviceCapability ( <a href="#">see page 120</a> ())
 <b>DAGetDeviceCapabilitiesHandle</b> ( <a href="#">see page 134</a> )	This function will provide a capabilite handle accoring to the gives device index. Call this function before you call AnalyseDeviceCapability ( <a href="#">see page 120</a> ()). <b>Direct access version.</b>
 <b>DAGetDeviceInformation</b> ( <a href="#">see page 135</a> )	This function reads the device information of a device from the internal device list. <b>Direct access version.</b>
 <b>DAGetDeviceInformationEx</b> ( <a href="#">see page 135</a> )	This function reads the device information of a device from the internal device list. <b>Direct access version.</b>
 <b>DAGetMaxBurnSpeed</b> ( <a href="#">see page 136</a> )	This function returns the max. wrting speed of the current device. <b>Direct access version.</b>
 <b>DAGetMaxReadSpeed</b> ( <a href="#">see page 136</a> )	This function reads the max. reading speed of the current device. <b>Direct access version.</b>
 <b>DAGetMediumFreedBld</b> ( <a href="#">see page 137</a> )	Use this function will receive the unique disc id for the audio disc. <b>Direct access version.</b>
 <b>DAGetMediumInformation</b> ( <a href="#">see page 137</a> )	This function returns the information about the current disc in the selected device. <b>Direct access version.</b>
 <b>DAGetPossibleBurnSpeeds</b> ( <a href="#">see page 138</a> )	This function returns a structure of the available burning speeds of the current device. <b>Direct access version.</b>
 <b>DAGetPossibleImageFormats</b> ( <a href="#">see page 138</a> )	This function will check what kind of image formats are possible to save from selected disc. <b>Direct access version.</b>
 <b>DAGetPossibleReadSpeeds</b> ( <a href="#">see page 139</a> )	This function returns a structure of the available reading speeds of the current device. <b>Direct access version.</b>
 <b>DAGetReadSpeed</b> ( <a href="#">see page 139</a> )	This function returns the reading speed of the current device. <b>Direct access version.</b>
 <b>DAGetSessionInformation</b> ( <a href="#">see page 140</a> )	This function returns the session information of the current disc in the current device. <b>Direct access version.</b>
 <b>DAGetTrackFormatEx</b> ( <a href="#">see page 140</a> )	This function returns the type of the disc track of the current medium of the given device. <b>Direct access version.</b>
 <b>DAGetTrackIndexes</b> ( <a href="#">see page 141</a> )	This function returns a list of sub indexes of the give audio track. <b>Direct access version.</b>
 <b>DAGetTrackInformation</b> ( <a href="#">see page 141</a> )	This function returns the track information of the current disc in the current device. <b>Direct access version.</b>

 DAGetTrackISRC (see page 142)	This function returns the ISRC code from the give audio track. <b>Direct access version.</b>
 DAGrabAudioTrack (see page 142)	The function will grab/save a audio track to a file. Tagging and different encoding types are available. <b>Direct access version.</b>
 DAImportFile (see page 143)	This function imports a file from a session to your project. <b>Direct access version.</b>
 DAImportFileEx (see page 143)	This function imports a file from a session to your project. <b>Direct access version.</b>
 DAIsDeviceReady (see page 144)	This function tests if the current device is ready. <b>Direct access version.</b>
 DALockMedium (see page 144)	This function will lock/unlock a disc inside a selected device. <b>Direct access version.</b>
 DAOpenDiskSession (see page 145)	This function opens a session on the selected disc/device. <b>Direct access version.</b>
 DAPlayAudioTrack (see page 145)	This function will play a audio track on the current disc or disc image. <b>Direct access version.</b>
 DAPrepare (see page 146)	This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables. <b>Direct access version.</b>
 DAReadCDText (see page 146)	This function will create a handle to the CD-Text information of the audio disc according to the give device index. <b>Direct access version.</b>
 DAReadFileContents (see page 147)	This function reads a segment of file from disc to a buffer. <b>Direct access version.</b>
 DAReadSectors (see page 147)	This function will read a given sector into a buffer. <b>Direct access version.</b>
 DASaveTrackToFile (see page 148)	This function saves a selected track to a file. <b>Direct access version.</b>
 DASetBurnSpeed (see page 148)	This function sets the writing speed of the current device. <b>Direct access version.</b>
 DASetReadSpeed (see page 149)	This function sets the reading speed of the current device. <b>Direct access version.</b>
 DASetRegionalCode (see page 149)	This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0. <b>Direct access version.</b>
 DAVerifyFile (see page 150)	This function verify a file of the medium against whether it is readable without any error. <b>Direct access version.</b>
 DelInitALL (see page 150)	This is function DelInitALL.
 DelInitialize (see page 151)	This function de-initializes the IsoSDK. Components of the IsoSDK will be released from memory.
 DeleteProject (see page 151)	This function deletes the current project.
 DirExists (see page 151)	This function will check whether a directory exists inside the project tree.
 EjectDevice (see page 152)	This function opens the current device (tray).
 EnableImageDevice (see page 152)	Enable or disable the internal ImageWriter device inside the device list.
 EnableMCNDisabling (see page 153)	This function will enable / disable the media change notification event on windows driven computers. Use this function to work with robotic systems.

	Erase ( <a href="#">see page 153</a> )	This function erases the disc in the current device.
	EraseDialog ( <a href="#">see page 154</a> )	This function starts the deletion process with the embedded dialog.
	EraseMpegByIndex ( <a href="#">see page 154</a> )	This function deletes a previously added MPEG file from a VCD or SVCD project.
	ForceDeInitialize ( <a href="#">see page 154</a> )	This is function ForceDeInitialize.
	GetActiveDevicesCount ( <a href="#">see page 155</a> )	This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.
	GetASPI ( <a href="#">see page 155</a> )	With this function you can get the information about the currently used ASPI layer.
	GetAudioFileSize ( <a href="#">see page 156</a> )	Use this function enumerate the final size of the compressed audio file
	GetAudioTags ( <a href="#">see page 156</a> )	This is function GetAudioTags.
	GetBootInfoEx ( <a href="#">see page 156</a> )	Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.
	GetBootVolumeInformation ( <a href="#">see page 157</a> )	Use this function to receive information about the boot file system on a existing disc or image.
	GetBurnDevice ( <a href="#">see page 157</a> )	This function returns the currently selected burn device. If you use multiple devices this function will return more than one device. Check the array with GetActiveDeviceCount()
	GetBurnDevices ( <a href="#">see page 158</a> )	This function returns an array with all selected burn devices.
	GetBurnDoneEventCallback ( <a href="#">see page 158</a> )	Hooks an already set BurnDoneEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetBurnFileEventCallback ( <a href="#">see page 159</a> )	Hooks an already set BurnFileEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetBurnSpeed ( <a href="#">see page 159</a> )	This function returns the writing speed of the selected device.
	GetCDTextDiskTagString ( <a href="#">see page 160</a> )	This function will get a specified CD-Text item from the created CD-Text handle of a audio disc.
	GetCDTextTrackTagString ( <a href="#">see page 160</a> )	This function will get a specified CD-Text track item from the created CD-Text handle of a audio disc.
	GetCompareFilesForArrangementEventCallback ( <a href="#">see page 161</a> )	Hooks an already set CompareFilesForArrangementEvent ( <a href="#">see page 237</a> ). Use this if you need the event in another dialog.
	GetCompressEncrypt ( <a href="#">see page 161</a> )	This function will get the parameters for internal compression and encryption.
	GetDeviceCapabilities ( <a href="#">see page 162</a> )	If you need to know the device capabilities use this function. <b>DEPRECATED! IS RELPACED BY:</b> AnalyseDeviceCapability ( <a href="#">see page 120</a> )()
	GetDeviceCapabilitiesHandle ( <a href="#">see page 162</a> )	This function will provide a capabilite handle accoring to the gives device index. Call this function before you call AnalyseDeviceCapability ( <a href="#">see page 120</a> )().
	GetDeviceInformation ( <a href="#">see page 163</a> )	This function reads the device information of a device from the internal device list.
	GetDeviceInformationEx ( <a href="#">see page 163</a> )	This function reads the device information of a device from the internal device list.
	GetDevices ( <a href="#">see page 164</a> )	This function returns a list of all available devices or a list of available devices with burning capabilities.
	GetDVDVideoOptions ( <a href="#">see page 164</a> )	This function gets the special options for Video DVD projects.
	GetEraseDoneEventCallback ( <a href="#">see page 165</a> )	Hooks an already set EraseDoneEvent ( <a href="#">see page 238</a> ). Use this if you need the event in another dialog
	GetErrorDeviceName ( <a href="#">see page 165</a> )	This function will return the name of the drive that reported the last error. This is helpful if you use multi-device burning.
	GetFileAllocationTable ( <a href="#">see page 165</a> )	Use this function to receive information of the file allocation table and the file extents on the disc according to the given file.

	GetFileEntry ( <a href="#">see page 166</a> )	This function will allow you to get all information about an already added file.
	GetFileTimeEx ( <a href="#">see page 166</a> )	Use this function to read the extended FileTimeEx data for the project types.
	GetFinalizeEventCallback ( <a href="#">see page 167</a> )	Hooks an already set FinalizeEvent ( <a href="#">see page 239</a> ). Use this if you need the event in another dialog.
	GetImagePath ( <a href="#">see page 167</a> )	This function will received the path to a opened image file.
	GetImageSize ( <a href="#">see page 168</a> )	This function returns the project size. The GetImageSize function is only available after the execution of the Prepare ( <a href="#">see page 194</a> ) function.
	GetInfoTextEventCallback ( <a href="#">see page 168</a> )	Hooks an already set InfoTextEvent ( <a href="#">see page 240</a> ). Use this if you need the event in another dialog.
	GetISOInfoEx ( <a href="#">see page 168</a> )	Use this function to read the extended ISO fields for Data CD/DVD/BD and ISO images. Available only in data projects.
	GetISOVolumeInformation ( <a href="#">see page 169</a> )	Use this function to receive information about the ISO filesystem on a existing disc or image.
	GetJobDoneEventCallback ( <a href="#">see page 169</a> )	Hooks an already set JobDoneEvent ( <a href="#">see page 241</a> ). Use this if you need the event in another dialog.
	GetLanguage ( <a href="#">see page 170</a> )	Use this function to determine the current language of the IsoSDK.
	GetMaxBurnSpeed ( <a href="#">see page 170</a> )	This function returns the max. writing speed of the current device.
	GetMaxReadSpeed ( <a href="#">see page 171</a> )	This function reads the max. reading speed of the current device.
	GetMediumFreedId ( <a href="#">see page 171</a> )	Use this function will receive the unique disc id for the audio disc.
	GetMediumInformation ( <a href="#">see page 172</a> )	This function returns the information about the current disc in the selected device.
	GetMpegCount ( <a href="#">see page 172</a> )	This function returns the number of added files from a VCD or SVCD project.
	GetNetworkDiskTagInt ( <a href="#">see page 172</a> )	This function will received a numeric value of a Cddb disc tag.
	GetNetworkDiskTagString ( <a href="#">see page 173</a> )	This function will received a string value of a Cddb disc tag.
	GetNetworkTrackTagInt ( <a href="#">see page 174</a> )	This function will received a numeric value of a Cddb track tag according to the given track.
	GetNetworkTrackTagString ( <a href="#">see page 174</a> )	This function will received a string value of a Cddb track tag according to the given track.
	GetNumberOfFiles ( <a href="#">see page 175</a> )	This function returns the number of files of the current disc in the current device.
	GetOptions ( <a href="#">see page 175</a> )	Use this function to read the burning options.
	GetPlayTime ( <a href="#">see page 176</a> )	This function returns the playtime of the given audio file in seconds.
	GetPossibleBurnSpeeds ( <a href="#">see page 176</a> )	This function returns a structure of the available burning speeds of the current device.
	GetPossibleImageFormats ( <a href="#">see page 177</a> )	This function will check what kind of image formats are possible to save from selected disc.
	GetPossibleReadSpeeds ( <a href="#">see page 177</a> )	This function returns a structure of the available reading speeds of the current device.
	GetPrecisePlayTime ( <a href="#">see page 178</a> )	This function returns the playtime of the given audio file in milliseconds.
	GetProcessEventCallback ( <a href="#">see page 178</a> )	Hooks an already set ProcessEvent ( <a href="#">see page 242</a> ). Use this if you need the event in another dialog.
	GetProjectType ( <a href="#">see page 178</a> )	This function returns the type of the current project.
	GetRAWDataEventCallback ( <a href="#">see page 179</a> )	Hooks an already set RAWDataEvent ( <a href="#">see page 242</a> ). Use this if you need the event in another dialog.

	GetReadDevice (see page 179)	This function returns the currently selected read device.
	GetReadSpeed (see page 180)	This function returns the reading speed of the current device.
	GetSessionInformation (see page 180)	This function returns the session information of the current disc in the current device.
	GetStartVerifyEventCallback (see page 181)	Hooks an already set StartVerifyEvent (see page 243). Use this if you need the event in another dialog.
	GetText (see page 181)	This function returns the text of a text ID according to your language setting; e.g. error code.
	GetTmpPath (see page 182)	To determine the current temporary path, call this function.
	GetTrackFormatEx (see page 182)	This function returns the type of the disc track of the current medium of the given device.
	GetTrackIndexes (see page 183)	This function returns a list of sub indexes of the give audio track.
	GetTrackInformation (see page 183)	This function returns the track information of the current disc in the current device.
	GetTrackISRC (see page 184)	This function returns the ISRC code from the give audio track.
	GetUDFOptions (see page 184)	Use this function to set the UDF options for UDF data projects.
	GetUDFOptionsEx (see page 185)	Use this function to read the UDF extended options for UDF data projects.
	GetUDFVolumeInformation (see page 185)	Use this function to receive information about the UDF file system on a existing disc or image.
	GetVerify (see page 185)	Use this function to get the afterburn verification state.
	GetVerifyDoneEventCallback (see page 186)	Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.
	GetVerifyErrorEventCallback (see page 186)	Hooks an already set VerifyDoneEvent (see page 244). Use this if you need the event in another dialog.
	GetVerifyFileEventCallback (see page 187)	Hooks an already set VerifyFileEvent (see page 245). Use this if you need the event in another dialog.
	GetVerifySectorEventCallback (see page 187)	Hooks an already set VerifySectorEvent (see page 245). Use this if you need the event in another dialog.
	GetWriteCDTextInUnicode (see page 187)	Use this function to read the CD-Text Unicode switch.
	GrabAudioTrack (see page 188)	The function will grab/save a audio track to a file. Tagging and different encoding types are available.
	ImportFile (see page 188)	This function imports a file from a session to your project.
	ImportFileEx (see page 189)	This function imports a file from a session to your project.
	Initialize (see page 189)	This function initializes the IsoSDK. The function must be called before you can use other functions.
	IsDeviceReady (see page 190)	This function tests if the current device is ready.
	IsValidVideoTsFolder (see page 190)	This function will check if a given path contains a valid VideoDVD structure with IFO, BUP and VOB Files.
	LoadBassPlugin (see page 191)	This function will load a given Bass plugin to the IsoSDK. Bass.dll have to be present. <a href="http://www.un4seen.com/">http://www.un4seen.com/</a> <b>NOTE:</b> The bass.dll and plugins are available as 32 and 64 bit.
	LockMedium (see page 191)	This function will lock/unlock a disc inside a selected device.
	MultiDeviceDialog (see page 192)	This function starts the multi-device selection dialog to burn on more than one device.
	OpenDirectory (see page 192)	This function opens a directory on the selected session.
	OpenDiskSession (see page 193)	This function opens a session on the selected disc/device.
	PlayAudioFile (see page 193)	Use this function to play the given file to the default output device.

	PlayAudioTrack ( <a href="#">see page 194</a> )	This function will play a audio track on the current disc or disc image.
	Prepare ( <a href="#">see page 194</a> )	This function prepares the data to be written. In this step the IsoSDK creates the file system and all tables.
	ReadCDText ( <a href="#">see page 194</a> )	This function will create a handle to the CD-Text information of the audio disc according to the give device index.
	ReadDirectory ( <a href="#">see page 195</a> )	This function returns a pointer to a structure representing the directory entry at the specified position.
	ReadFileContents ( <a href="#">see page 195</a> )	This function reads a segment of file from disc to a buffer.
	ReadSectors ( <a href="#">see page 196</a> )	This function will read a given sector into a buffer.
	ReleaseDeviceCapabilities ( <a href="#">see page 196</a> )	Release a created capabilite handle to clean up the memory.
	RemoveBurnDevice ( <a href="#">see page 197</a> )	This function will remove a device from the multi burn device list. This is important for multi-device burning.
	RemoveDir ( <a href="#">see page 197</a> )	This function removes a directory from the current project.
	RemoveFile ( <a href="#">see page 198</a> )	
	RenameDir ( <a href="#">see page 198</a> )	This function rename / move an directory in the project.
	RenameFile ( <a href="#">see page 198</a> )	This function rename / move a file in the project.
	RescanDevices ( <a href="#">see page 199</a> )	This function will rescan the internal device list. This is useful to detect new connected devices like USB and FireWire.
	ResetCallbacks ( <a href="#">see page 199</a> )	This function will reset all initialized callbacks to your project.
	SaveLogToFile ( <a href="#">see page 200</a> )	This function will save the automatically generated burning log data to a file.
	SaveOptionsToFile ( <a href="#">see page 200</a> )	This function will save the burn settings/ options to a XML File.
	SaveTrackToFile ( <a href="#">see page 201</a> )	This function saves a selected track to a file.
	SetAddFileEventCallback ( <a href="#">see page 201</a> )	Specifies an own function as a callback of the type AddFileEvent ( <a href="#">see page 235</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetASPI ( <a href="#">see page 202</a> )	If you want to use the internal ASPI layer after the initialization you use this function to switch from the external to the internal ASPI layer et vice versa .
	SetAudioDecodeDoneEventCallback ( <a href="#">see page 203</a> )	Specifies an own function as a callback of the type AudioDecodeDoneEvent ( <a href="#">see page 236</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetAudioDecoderEventCallback ( <a href="#">see page 203</a> )	Specifies an own function as a callback of the type AudioDecoderEvent ( <a href="#">see page 236</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetAudioFileProperty ( <a href="#">see page 204</a> )	The function sets the audio properties for the specified audio file in the project or for the disc. Supported projects: Audio, Mixed Mode.
	SetBootInfoEx ( <a href="#">see page 204</a> )	Use this function to set the extended boot image options for Data CD/DVD and ISO images. Available only in data projects.
	SetBurnDevice ( <a href="#">see page 205</a> )	This function sets the current device. This drive will be used for the burning process.
	SetBurnDoneEventCallback ( <a href="#">see page 205</a> )	Specifies an own function as a callback of the type BurnDoneEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetBurnFileEventCallback ( <a href="#">see page 206</a> )	Specifies an own function as a callback of the type BurnFileEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetBurnSpeed ( <a href="#">see page 206</a> )	This function sets the writing speed of the current device.
	SetCompareFilesForArrangementEventCallback ( <a href="#">see page 207</a> )	Specifies an own function as a callback of the type CompareFilesForArrangementEvent ( <a href="#">see page 237</a> ). This function will be called every time the event triggers inside the IsoSDK.

	<a href="#">SetCompressEncrypt</a> (see page 208)	This function will set the parameters for internal compression and encryption.
	<a href="#">SetCreateDirEventCallback</a> (see page 208)	Specifies an own function as a callback of the type CreateDirEvent (see page 238). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetDVDVideoOptions</a> (see page 209)	This function sets the special options for VideoDVD projects.
	<a href="#">SetEraseDoneEventCallback</a> (see page 209)	Specifies an own function as a callback of the type EraseDoneEvent (see page 238). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetFileAttr</a> (see page 210)	This function set the attributes to a file.
<b>new</b>	<a href="#">SetFileTimeEx</a> (see page 210)	<p>This function is used to manage the custom date / time values for files, directory and file system entries. With this values you can overwrite the common date / time values of the files.</p> <p>You can use the system time or own time values. This date / time value is set global but you can overwrite each file value with SetFileTimes (see page 211) method.</p> <p>This settings will not changed old added files / directories only those that get added after calling this. Will get reset with define a new project.</p>
	<a href="#">SetFileTimes</a> (see page 211)	This function will set the custom dates to a file.
	<a href="#">SetFileUserParam</a> (see page 211)	This is a data value you can set yourself according to your needs. You can pass a integer or structure to the file that stores extra data.
	<a href="#">SetFinalizeEventCallback</a> (see page 212)	Specifies an own function as a callback of the type FinalizeEvent (see page 239). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetFXApp</a> (see page 212)	Set the Fox-Toolkit App handle to the SDK.
	<a href="#">SetGetTextEventCallback</a> (see page 213)	Specifies an own function as a callback of the type GetTextEvent (see page 239). This function will be called every time the event triggers inside the IsoSDK.
<b>new</b>	<a href="#">SetIgnoreFileExist</a> (see page 213)	The IsoSDK will check about already added files while AddFile (see page 119) and AddDir (see page 118) function. If you have industrial workflows with thousands of files you can disable this check with this function.
	<a href="#">SetImagePath</a> (see page 214)	The function sets the path to the image (*.iso) file you want to burn using the 'Imagewriter'.
	<a href="#">SetInfoTextEventCallback</a> (see page 214)	Specifies an own function as a callback of the type InfoTextEvent (see page 240). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetISOInfoEx</a> (see page 215)	Use this function to set the extended ISO Fields for Data CD/DVD and ISO images. Available only in data projects.
	<a href="#">SetJobDoneEventCallback</a> (see page 215)	Specifies an own function as a callback of the type JobDoneEvent (see page 241). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetLanguage</a> (see page 216)	This function sets the language of the IsoSDK. The following resources are effected: error codes, message codes and GUI codes.
	<a href="#">SetOptions</a> (see page 217)	Use this function to set the burning options.
	<a href="#">SetOptionsFromFile</a> (see page 217)	This function will set the burn settings/ options from a previously stored XML file.
	<a href="#">SetProcessEventCallback</a> (see page 217)	Specifies an own function as a callback of the type ProcessEvent (see page 242). This function will be called every time the event triggers inside the IsoSDK.
	<a href="#">SetRAWDataEventCallback</a> (see page 218)	Specifies an own function as a callback of the type RAWDataEvent (see page 242). This function will be called every time the event triggers inside the IsoSDK.

	SetRAWStructure ( <a href="#">see page 219</a> )	The function sets the structure of the disc and the format of the input data for a RAW project.
	SetReadDevice ( <a href="#">see page 219</a> )	This function sets the reading speed of the current read device.
	SetReadSpeed ( <a href="#">see page 219</a> )	This function sets the reading speed of the current device.
	SetRegionalCode ( <a href="#">see page 220</a> )	This function will change the region code of a device. You can set 1-6. 7 and 8 are not allowed to set. You also can not set back to "unset" mode 0.
	SetRemoveFileEventCallback ( <a href="#">see page 221</a> )	Specifies an own function as a callback of the type RemoveFileEvent ( <a href="#">see page 243</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetStartVerifyEventCallback ( <a href="#">see page 221</a> )	Specifies an own function as a callback of the type StartVerifyEvent ( <a href="#">see page 243</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetTmpPath ( <a href="#">see page 222</a> )	With the help of this function the temporary folder is set. A previously set path will be overwritten.
	SetUDFOptions ( <a href="#">see page 222</a> )	Use this function to set the UDF options for UDF data projects.
	SetUDFOptionsEx ( <a href="#">see page 223</a> )	Use this function to set the UDF extended options for UDF data projects. Insert additional UDF settings with this function.
	SetVCDKeyHandler ( <a href="#">see page 223</a> )	This function sets which item will be shown next when particular key is pressed.
	SetVCDTimeOutHandler ( <a href="#">see page 224</a> )	This function sets transition timeout between items.
	SetVerify ( <a href="#">see page 224</a> )	Use this function to set the after burn verification.
	SetVerifyDoneEventCallback ( <a href="#">see page 224</a> )	Specifies an own function as a callback of the type VerifyDoneEvent ( <a href="#">see page 244</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVerifyErrorEventCallback ( <a href="#">see page 225</a> )	Specifies an own function as a callback of the type VerifyErrorEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the SDK.
	SetVerifyFileEventCallback ( <a href="#">see page 226</a> )	Specifies an own function as a callback of the type VerifyFileEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVerifySectorEventCallback ( <a href="#">see page 226</a> )	Specifies an own function as a callback of the type VerifySectorEvent ( <a href="#">see page 245</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVideoScanDoneEventCallback ( <a href="#">see page 227</a> )	Specifies an own function as a callback of the type VideoScanDoneEvent ( <a href="#">see page 246</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetVideoScannerEventCallback ( <a href="#">see page 228</a> )	Specifies an own function as a callback of the type VideoScannerEvent ( <a href="#">see page 246</a> ). This function will be called every time the event triggers inside the IsoSDK.
	SetWriteCDTextInUnicode ( <a href="#">see page 228</a> )	Use this function to set the CD-Text Unicode switch.
	StopMpegAction ( <a href="#">see page 229</a> )	This function interrupts the current scanning process of an added video file to a VCD or SVCD project.
	TagsFromNetworkDialog ( <a href="#">see page 229</a> )	This function will create a handle to the CDDDB information of the audio disc according to the give device index.
	VerifyFile ( <a href="#">see page 230</a> )	This function verify a file of the medium against whether it is readable without any error.

## Macros

Name	Description
<code>__ISOSDK_EXPORT_H__</code> ( <a href="#">see page 306</a> )	This is macro <code>__ISOSDK_EXPORT_H__</code> .
<code>BS_CALL</code> ( <a href="#">see page 317</a> )	This is macro <code>BS_CALL</code> .

## 1.6.4 IsoSDKUnicode.h

This is file IsoSDKUnicode.h.

### Macros

Name	Description
_FindData_t ( <a href="#">see page 307</a> )	This is macro _FindData_t.
_FindFirst ( <a href="#">see page 307</a> )	This is macro _FindFirst.
FOpen ( <a href="#">see page 512</a> )	This is macro FOpen.
SPrintf ( <a href="#">see page 514</a> )	This is macro SPrintf.
StrCmp ( <a href="#">see page 514</a> )	This is macro StrCmp.
StrCpy ( <a href="#">see page 514</a> )	This is macro StrCpy.
StrLen ( <a href="#">see page 514</a> )	This is macro StrLen.
StrnSet ( <a href="#">see page 515</a> )	This is macro StrnSet.
VsPrintf ( <a href="#">see page 516</a> )	This is macro VsPrintf.

# Index

—  
 \_\_\_stat64 305  
 \_\_\_stat64 macro 305  
 \_\_BS\_LIBRARY\_H\_\_ 305  
 \_\_BS\_LIBRARY\_H\_\_ macro 305  
 \_\_ISOSDK\_DEFINITIONS\_H\_\_ 306  
 \_\_ISOSDK\_DEFINITIONS\_H\_\_ macro 306  
 \_\_ISOSDK\_EXPORT\_H\_\_ 306  
 \_\_ISOSDK\_EXPORT\_H\_\_ macro 306  
 \_\_max 306  
 \_\_max macro 306  
 \_\_min 306  
 \_\_min macro 306  
 \_FindData\_t 307  
 \_FindData\_t macro 307  
 \_FindFirst 307  
 \_FindFirst macro 307  
 \_FoxEntries 231  
 \_FoxEntries structure 231  
 \_FoxIndecies 231  
 \_FoxIndecies structure 231  
 \_MAX\_PATH 307  
 \_MAX\_PATH macro 307

## A

Abort 118  
 Abort function 118  
 AddBurnDevice 118  
 AddBurnDevice function 118  
 AddDir 118  
 AddDir function 118  
 AddFile 119  
 AddFile function 119  
 AddFileEvent 235  
 AddFileEvent type 235  
 AddFileEx 120  
 AddFileEx function 120  
 AnalyseDeviceCapability 120  
 AnalyseDeviceCapability function 120

AudioDecodeDoneEvent 236  
 AudioDecodeDoneEvent type 236  
 AudioDecoderEvent 236  
 AudioDecoderEvent type 236  
 AudioFileStop 121  
 AudioFileStop function 121  
**B**  
 BOOL2bool 307  
 BOOL2bool macro 307  
 BS\_ANALOG\_AUDIO\_PLAYBACK 308  
 BS\_ANALOG\_AUDIO\_PLAYBACK macro 308  
 BS\_API 308  
 BS\_API macro 308  
 BS\_AR\_16TO9\_DISPLAY 308  
 BS\_AR\_16TO9\_DISPLAY macro 308  
 BS\_AR\_221TO2\_DISPLAY 308  
 BS\_AR\_221TO2\_DISPLAY macro 308  
 BS\_AR\_4TO3\_DISPLAY 309  
 BS\_AR\_4TO3\_DISPLAY macro 309  
 BS\_AR\_SQUARE\_PIXELS 309  
 BS\_AR\_SQUARE\_PIXELS macro 309  
 BS\_AR\_UNKNOWN 309  
 BS\_AR\_UNKNOWN macro 309  
 BS\_ASPI\_FROGASPI 309  
 BS\_ASPI\_FROGASPI macro 309  
 BS\_ASPI\_INTERNAL 310  
 BS\_ASPI\_INTERNAL macro 310  
 BS\_ASPI\_WNASPI 310  
 BS\_ASPI\_WNASPI macro 310  
 BS\_AUDIO\_MP3 310  
 BS\_AUDIO\_MP3 macro 310  
 BS\_AUDIO\_NO 310  
 BS\_AUDIO\_NO macro 310  
 BS\_AUDIO\_OGG 311  
 BS\_AUDIO\_OGG macro 311  
 BS\_AUDIO\_PCM 311  
 BS\_AUDIO\_PCM macro 311  
 BS\_BARCODE\_READ 311  
 BS\_BARCODE\_READ macro 311  
 BS\_BLURAY\_R 311  
 BS\_BLURAY\_R macro 311

BS_BLURAY_R_RRM	312	BS_CALL	317
BS_BLURAY_R_RRM macro	312	BS_CALL macro	317
BS_BLURAY_RE	312	BS_CD_R	317
BS_BLURAY_RE macro	312	BS_CD_R macro	317
BS_BLURAY_ROM	312	BS_CD_ROM	317
BS_BLURAY_ROM macro	312	BS_CD_ROM macro	317
BS_BOOL	312	BS_CD_RW	318
BS_BOOL macro	312	BS_CD_RW macro	318
BS_BOOL2bool	313	BS_CD_TEXT_READ	318
BS_BOOL2bool macro	313	BS_CD_TEXT_READ macro	318
BS_BT_AVERAGE	313	BS_CD_TEXT_WRITE	318
BS_BT_AVERAGE macro	313	BS_CD_TEXT_WRITE macro	318
BS_BT_CONSTANT	313	BS_CDDA_COMMANDS	318
BS_BT_CONSTANT macro	313	BS_CDDA_COMMANDS macro	318
BS_BT_VARIABLE	313	BS_CDDA_STREAM_IS_ACCURATE	319
BS_BT_VARIABLE macro	313	BS_CDDA_STREAM_IS_ACCURATE macro	319
BS_BURN_DEVICE	314	BS_CDTCI_ARRANGER	319
BS_BURN_DEVICE macro	314	BS_CDTCI_ARRANGER macro	319
BS_BURNER_AUDIO	314	BS_CDTCI_COMPOSER	319
BS_BURNER_AUDIO macro	314	BS_CDTCI_COMPOSER macro	319
BS_BURNER_BLURAY	314	BS_CDTCI_MESSAGE	319
BS_BURNER_BLURAY macro	314	BS_CDTCI_MESSAGE macro	319
BS_BURNER_CUE	314	BS_CDTCI_PERFORMER	320
BS_BURNER_CUE macro	314	BS_CDTCI_PERFORMER macro	320
BS_BURNER_DATA	315	BS_CDTCI_SONG_WRITER	320
BS_BURNER_DATA macro	315	BS_CDTCI_SONG_WRITER macro	320
BS_BURNER_ISOUDF	315	BS_CDTCI_TITLE	320
BS_BURNER_ISOUDF macro	315	BS_CDTCI_TITLE macro	320
BS_BURNER_MIXEDMODE	315	BS_CHANGER_SIDE_CHANGE_CAPABLE	320
BS_BURNER_MIXEDMODE macro	315	BS_CHANGER_SIDE_CHANGE_CAPABLE macro	320
BS_BURNER_RAW	315	BS_CHANGER_SOFTWARE_SLOT_SELECTION	321
BS_BURNER_RAW macro	315	BS_CHANGER_SOFTWARE_SLOT_SELECTION macro	321
BS_BURNER_SVCD	316	BS_CHANGER_SUPPORTS_DISC_PRESENT	321
BS_BURNER_SVCD macro	316	BS_CHANGER_SUPPORTS_DISC_PRESENT macro	321
BS_BURNER_UDFDVD	316	BS_COMPOSITE_AUDIO_AND_VIDEO	321
BS_BURNER_UDFDVD macro	316	BS_COMPOSITE_AUDIO_AND_VIDEO macro	321
BS_BURNER_VCD	316	BS_COMPRESSED_SIGNATURE	321
BS_BURNER_VCD macro	316	BS_COMPRESSED_SIGNATURE macro	321
BS_BURNER_VIDEODVD	316	BS_COMPRESSED_SIGNATURE_FOR_DRIVER	322
BS_BURNER_VIDEODVD macro	316	BS_COMPRESSED_SIGNATURE_FOR_DRIVER macro	322
BS_C2_POINTERS	317	BS_CONTINUE_LAST_SESSION	322
BS_C2_POINTERS macro	317	BS_CONTINUE_LAST_SESSION macro	322

BS_CONTINUE_NO_SESSION 322	BS_DVD_RAM 327
BS_CONTINUE_NO_SESSION macro 322	BS_DVD_RAM macro 327
BS_CPRMAUTH 322	BS_DVD_RDL_PLUS 328
BS_CPRMAUTH macro 322	BS_DVD_RDL_PLUS macro 328
BS_CURRENT_DEVICE 323	BS_DVD_ROM 328
BS_CURRENT_DEVICE macro 323	BS_DVD_ROM macro 328
BS.DAO_16 323	BS_DVD_RW 328
BS.DAO_16 macro 323	BS_DVD_RW macro 328
BS.DAO_96_PACK 323	BS_DVD_RW_RO 328
BS.DAO_96_PACK macro 323	BS_DVD_RW_RO macro 328
BS.DAO_96_RAW 323	BS_DVD_RW_SR 329
BS.DAO_96_RAW macro 323	BS_DVD_RW_SR macro 329
BS.DAO_RAW 324	BS_DVD_RWDL_PLUS 329
BS.DAO_RAW macro 324	BS_DVD_RWDL_PLUS macro 329
BS.DDCD.R 324	BS_EJECT_INDIVIDUAL_OR_MAGAZINE 329
BS.DDCD.R macro 324	BS_EJECT_INDIVIDUAL_OR_MAGAZINE macro 329
BS.DDCD.ROM 324	BS.EMT.BD 329
BS.DDCD.ROM macro 324	BS.EMT.BD macro 329
BS.DDCD.RW 324	BS.EMT.CD_AUDIO 330
BS.DDCD.RW macro 324	BS.EMT.CD_AUDIO macro 330
BS.DEFAULT_LANGUAGE_FILE 325	BS.EMT.CD_ENHANCED 330
BS.DEFAULT_LANGUAGE_FILE macro 325	BS.EMT.CD_ENHANCED macro 330
BS.DEFAULT_REGISTRY_KEY 325	BS.EMT.CD_MIXED_MODE 330
BS.DEFAULT_REGISTRY_KEY macro 325	BS.EMT.CD_MIXED_MODE macro 330
BS.DEFAULT_REGISTRY_PATH 325	BS.EMT.CD_MULTISESSION 330
BS.DEFAULT_REGISTRY_PATH macro 325	BS.EMT.CD_MULTISESSION macro 330
BS.DEFFECTMANAGEMENT 325	BS.EMT.CD.ROM 331
BS.DEFFECTMANAGEMENT macro 325	BS.EMT.CD.ROM macro 331
BS.DIGITAL_PORT_1 326	BS.EMT.CD.ROM_XA 331
BS.DIGITAL_PORT_1 macro 326	BS.EMT.CD.ROM_XA macro 331
BS.DIGITAL_PORT_2 326	BS.EMT.DVD 331
BS.DIGITAL_PORT_2 macro 326	BS.EMT.DVD macro 331
BS.DONT_SAVE_PATH 326	BS.EMT.HDDVD 331
BS.DONT_SAVE_PATH macro 326	BS.EMT.HDDVD macro 331
BS.DVD_MRDL 326	BS.ET.AAC 332
BS.DVD_MRDL macro 326	BS.ET.AAC macro 332
BS.DVD_PLUSR 327	BS.ET.FLAC 332
BS.DVD_PLUSR macro 327	BS.ET.FLAC macro 332
BS.DVD_PLUSRW 327	BS.ET.MP3 332
BS.DVD_PLUSRW macro 327	BS.ET.MP3 macro 332
BS.DVD_R 327	BS.ET.MP4 332
BS.DVD_R macro 327	BS.ET.MP4 macro 332

BS_ET_OGG 333	BS_GUI_ERROR_01 338
BS_ET_OGG macro 333	BS_GUI_ERROR_01 macro 338
BS_ET_OPUS 333	BS_GUI_ERROR_02 338
BS_ET_OPUS macro 333	BS_GUI_ERROR_02 macro 338
BS_ET_WMA 333	BS_GUI_RESOURCE_01 338
BS_ET_WMA macro 333	BS_GUI_RESOURCE_01 macro 338
BS_FA_ADVANCED_HIDDEN 333	BS_GUI_RESOURCE_02 339
BS_FA_ADVANCED_HIDDEN macro 333	BS_GUI_RESOURCE_02 macro 339
BS_FA_ALL 334	BS_GUI_RESOURCE_03 339
BS_FA_ALL macro 334	BS_GUI_RESOURCE_03 macro 339
BS_FA_ARCHIVE 334	BS_GUI_RESOURCE_04 339
BS_FA_ARCHIVE macro 334	BS_GUI_RESOURCE_04 macro 339
BS_FA_DIRECTORY 334	BS_GUI_RESOURCE_05 339
BS_FA_DIRECTORY macro 334	BS_GUI_RESOURCE_05 macro 339
BS_FA_HIDDEN 334	BS_GUI_RESOURCE_06 340
BS_FA_HIDDEN macro 334	BS_GUI_RESOURCE_06 macro 340
BS_FA_READONLY 335	BS_GUI_RESOURCE_07 340
BS_FA_READONLY macro 335	BS_GUI_RESOURCE_07 macro 340
BS_FA_SYSTEM 335	BS_GUI_RESOURCE_08 340
BS_FA_SYSTEM macro 335	BS_GUI_RESOURCE_08 macro 340
BS_FALSE 335	BS_GUI_RESOURCE_09 340
BS_FALSE macro 335	BS_GUI_RESOURCE_09 macro 340
BS_FF_BIN 335	BS_GUI_RESOURCE_10 341
BS_FF_BIN macro 335	BS_GUI_RESOURCE_10 macro 341
BS_FF_ISO 336	BS_GUI_RESOURCE_11 341
BS_FF_ISO macro 336	BS_GUI_RESOURCE_11 macro 341
BS_FF_MPEG 336	BS_GUI_RESOURCE_12 341
BS_FF_MPEG macro 336	BS_GUI_RESOURCE_12 macro 341
BS_FF_WAVE 336	BS_GUI_RESOURCE_13 342
BS_FF_WAVE macro 336	BS_GUI_RESOURCE_13 macro 342
BS_FS_BOOTABLE 336	BS_GUI_RESOURCE_14 342
BS_FS_BOOTABLE macro 336	BS_GUI_RESOURCE_14 macro 342
BS_FS_ISO9660 337	BS_GUI_RESOURCE_15 342
BS_FS_ISO9660 macro 337	BS_GUI_RESOURCE_15 macro 342
BS_FS_JOLIET 337	BS_GUI_RESOURCE_16 342
BS_FS_JOLIET macro 337	BS_GUI_RESOURCE_16 macro 342
BS_FS_ROCKRIDGE 337	BS_GUI_RESOURCE_17 343
BS_FS_ROCKRIDGE macro 337	BS_GUI_RESOURCE_17 macro 343
BS_FS_UDF 337	BS_GUI_RESOURCE_18 343
BS_FS_UDF macro 337	BS_GUI_RESOURCE_18 macro 343
BS_FS_UNKNOWN 338	BS_GUI_RESOURCE_19 343
BS_FS_UNKNOWN macro 338	BS_GUI_RESOURCE_19 macro 343

BS_GUI_RESOURCE_20	343	BS_GUI_RESOURCE_41	349		
BS_GUI_RESOURCE_20	macro	343	BS_GUI_RESOURCE_41	macro	349
BS_GUI_RESOURCE_21	344	BS_GUI_RESOURCE_42	349		
BS_GUI_RESOURCE_21	macro	344	BS_GUI_RESOURCE_42	macro	349
BS_GUI_RESOURCE_22	344	BS_GUI_RESOURCE_43	350		
BS_GUI_RESOURCE_22	macro	344	BS_GUI_RESOURCE_43	macro	350
BS_GUI_RESOURCE_23	344	BS_GUI_RESOURCE_44	350		
BS_GUI_RESOURCE_23	macro	344	BS_GUI_RESOURCE_44	macro	350
BS_GUI_RESOURCE_24	345	BS_GUI_RESOURCE_45	350		
BS_GUI_RESOURCE_24	macro	345	BS_GUI_RESOURCE_45	macro	350
BS_GUI_RESOURCE_25	345	BS_GUI_RESOURCE_46	350		
BS_GUI_RESOURCE_25	macro	345	BS_GUI_RESOURCE_46	macro	350
BS_GUI_RESOURCE_26	345	BS_GUI_RESOURCE_47	351		
BS_GUI_RESOURCE_26	macro	345	BS_GUI_RESOURCE_47	macro	351
BS_GUI_RESOURCE_27	345	BS_GUI_RESOURCE_48	351		
BS_GUI_RESOURCE_27	macro	345	BS_GUI_RESOURCE_48	macro	351
BS_GUI_RESOURCE_28	346	BS_GUI_RESOURCE_49	351		
BS_GUI_RESOURCE_28	macro	346	BS_GUI_RESOURCE_49	macro	351
BS_GUI_RESOURCE_29	346	BS_GUI_RESOURCE_50	352		
BS_GUI_RESOURCE_29	macro	346	BS_GUI_RESOURCE_50	macro	352
BS_GUI_RESOURCE_30	346	BS_GUI_RESOURCE_51	352		
BS_GUI_RESOURCE_30	macro	346	BS_GUI_RESOURCE_51	macro	352
BS_GUI_RESOURCE_31	346	BS_GUI_RESOURCE_52	352		
BS_GUI_RESOURCE_31	macro	346	BS_GUI_RESOURCE_52	macro	352
BS_GUI_RESOURCE_32	347	BS_GUI_RESOURCE_53	352		
BS_GUI_RESOURCE_32	macro	347	BS_GUI_RESOURCE_53	macro	352
BS_GUI_RESOURCE_33	347	BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLU MN	353		
BS_GUI_RESOURCE_33	macro	347	BS_GUI_RESOURCE_INTERNET_DIALOG_ARTIST_COLU MN	macro	353
BS_GUI_RESOURCE_34	347	BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_C OLUMN	353		
BS_GUI_RESOURCE_34	macro	347	BS_GUI_RESOURCE_INTERNET_DIALOG_CATEGORY_C OLUMN	macro	353
BS_GUI_RESOURCE_35	347	BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLU MN	353		
BS_GUI_RESOURCE_35	macro	347	BS_GUI_RESOURCE_INTERNET_DIALOG_DISCID_COLU MN	macro	353
BS_GUI_RESOURCE_36	348	BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABE L	354		
BS_GUI_RESOURCE_36	macro	348			
BS_GUI_RESOURCE_37	348				
BS_GUI_RESOURCE_37	macro	348			
BS_GUI_RESOURCE_38	348				
BS_GUI_RESOURCE_38	macro	348			
BS_GUI_RESOURCE_39	349				
BS_GUI_RESOURCE_39	macro	349			
BS_GUI_RESOURCE_40	349				
BS_GUI_RESOURCE_40	macro	349			

BS_GUI_RESOURCE_INTERNET_DIALOG_STATUS_LABEL macro	354	BS_IMPOPTS_COMMON macro	359
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE	354	BS_IMPOPTS_DECRYPT	359
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE macro	354	BS_IMPOPTS_DECRYPT macro	359
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN	354	BS_IMPOPTS_EX	359
BS_GUI_RESOURCE_INTERNET_DIALOG_TITLE_COLUMN macro	354	BS_IMPOPTS_EX macro	359
BS_HD_DVD_R	355	BS_IMPOPTS_UNCOMPRESS	359
BS_HD_DVD_R macro	355	BS_IMPOPTS_UNCOMPRESS macro	359
BS_HD_DVD_R_DL	355	BS_INVALID_HANDLE	360
BS_HD_DVD_R_DL macro	355	BS_INVALID_HANDLE macro	360
BS_HD_DVD_RAM	355	BS_INVALID_TAG_HANDLE	360
BS_HD_DVD_RAM macro	355	BS_ISO_LEVEL_1	360
BS_HD_DVD_ROM	355	BS_ISO_LEVEL_1 macro	360
BS_HD_DVD_ROM macro	355	BS_ISO_LEVEL_2	360
BS_HD_DVD_RW	356	BS_ISO_LEVEL_2 macro	360
BS_HD_DVD_RW macro	356	BS_ISO_LEVEL_3	361
BS_HD_DVD_RW_DL	356	BS_ISO_LEVEL_3 macro	361
BS_HD_DVD_RW_DL macro	356	BS_ISO_LEVEL_ROMEO	361
BS_IL_HIGH_DEBUG	356	BS_ISO_LEVEL_ROMEO macro	361
BS_IL_HIGH_DEBUG macro	356	BS_ISRC_READ	361
BS_IL_INFO	356	BS_ISRC_READ macro	361
BS_IL_INFO macro	356	BS_LABELFLASH	362
BS_IL_LOW_DEBUG	357	BS_LABELFLASH macro	362
BS_IL_LOW_DEBUG macro	357	BS_LAYER_JUMP_RECORDING	362
BS_IL_MEDIUM_DEBUG	357	BS_LAYER_JUMP_RECORDING macro	362
BS_IL_MEDIUM_DEBUG macro	357	BS_LIGHTSCRIBE	362
BS_ILE_TOO_BIG_FILE	357	BS_LIGHTSCRIBE macro	362
BS_ILE_TOO_BIG_FILE macro	357	BS_LOCK_MEDIA	362
BS_ILE_TOO_LONG_DIRECTORY_NESTING	357	BS_LOCK_MEDIA macro	362
BS_ILE_TOO_LONG_DIRECTORY_NESTING macro	357	BS_LOCK_STATE	363
BS_IMG_BIN	358	BS_LOCK_STATE macro	363
BS_IMG_BIN macro	358	BS_LP_BOOL	363
BS_IMG_ISO	358	BS_LP_BOOL macro	363
BS_IMG_ISO macro	358	BS_LS_COMPLETE_SESSION	363
BS_IMGTASK_CREATE	358	BS_LS_COMPLETE_SESSION macro	363
BS_IMGTASK_CREATE macro	358	BS_LS_DAMAGED_SESSION	363
BS_IMGTASK_VERIFY	358	BS_LS_DAMAGED_SESSION macro	363
BS_IMGTASK_VERIFY macro	358	BS_LS_EMPTY_SESSION	364
BS_IMPOPTS_COMMON	359	BS_LS_EMPTY_SESSION macro	364
		BS_LS_INCOMPLETE_SESSION	364
		BS_LS_INCOMPLETE_SESSION macro	364
		BS_MAX_SPEED	364

BS_MAX_SPEED macro 364	BS_RDT_SUBCH_PW macro 369
BS_METHOD_2_ADDRESSING_FIXED_PACKETS 364	BS_RDT_SUBCH_RW 370
BS_METHOD_2_ADDRESSING_FIXED_PACKETS macro 364	BS_RDT_SUBCH_RW macro 370
BS_MODE2_FORM1_READ 365	BS_RDT_SUBHEADERS 370
BS_MODE2_FORM1_READ macro 365	BS_RDT_SUBHEADERS macro 370
BS_MODE2_FORM2_READ 365	BS_RDT_SYNC_HEADER 370
BS_MODE2_FORM2_READ macro 365	BS_RDT_SYNC_HEADER macro 370
BS_MS_COMPLETE_DISK 365	BS_READ_BLURAY_R 370
BS_MS_COMPLETE_DISK macro 365	BS_READ_BLURAY_R macro 370
BS_MS_EMPTY_DISK 365	BS_READ_BLURAY_R_XL 371
BS_MS_EMPTY_DISK macro 365	BS_READ_BLURAY_R_XL macro 371
BS_MS_INCOMPLETE_DISK 366	BS_READ_BLURAY_RE 371
BS_MS_INCOMPLETE_DISK macro 366	BS_READ_BLURAY_RE macro 371
BS_MS_OTHER 366	BS_READ_BLURAY_RE_XL 371
BS_MS_OTHER macro 366	BS_READ_BLURAY_ROM 371
BS_MULTI_ERROR_01 366	BS_READ_BLURAY_ROM macro 371
BS_MULTI_ERROR_01 macro 366	BS_READ_CDR 372
BS_MULTI_ERROR_02 366	BS_READ_CDR macro 372
BS_MULTI_ERROR_02 macro 366	BS_READ_CDRW 372
BS_MULTISESSION 367	BS_READ_CDRW macro 372
BS_MULTISESSION macro 367	BS_READ_CDRW_CAV 372
BS_NDEF 367	BS_READ_CDRW_CAV macro 372
BS_NDEF macro 367	BS_READ_DEVICE 372
BS_PACKET_WRITE 367	BS_READ_DEVICE macro 372
BS_PACKET_WRITE macro 367	BS_READ_DVD 373
BS_PARENTDIR_ONLY 367	BS_READ_DVD macro 373
BS_PARENTDIR_ONLY macro 367	BS_READ_DVD_DL 373
BS_PREVENT_JUMPER 368	BS_READ_DVD_DL macro 373
BS_PREVENT_JUMPER macro 368	BS_READ_DVD_MRDL 373
BS_R_W_SUBCHANNELS_DEINT_AND_CORR 368	BS_READ_DVD_MRDL macro 373
BS_R_W_SUBCHANNELS_DEINT_AND_CORR macro 368	BS_READ_DVD_RDL_PLUS 373
BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ 368	BS_READ_DVD_RDL_PLUS macro 373
BS_R_W_SUBCHANNELS_IN_LEAD_IN_READ macro 368	BS_READ_DVD_RWDL_PLUS 374
BS_R_W_SUBCHANNELS_READ 368	BS_READ_DVD_RWDL_PLUS macro 374
BS_R_W_SUBCHANNELS_READ macro 368	BS_READ_DVDR 374
BS_RDT_DATA 369	BS_READ_DVDR macro 374
BS_RDT_DATA macro 369	BS_READ_DVDR_PLUS 374
BS_RDT_EDC_ECC 369	BS_READ_DVDR_PLUS macro 374
BS_RDT_EDC_ECC macro 369	BS_READ_DVDRAM 374
BS_RDT_SUBCH_PQ 369	BS_READ_DVDRAM macro 374
BS_RDT_SUBCH_PQ macro 369	BS_READ_DVDRW 375
BS_RDT_SUBCH_PW 369	

BS_READ_DVDRW macro 375	BS_SCSI_ERROR_009 macro 380
BS_READ_DVDRW_PLUS 375	BS_SCSI_ERROR_01 380
BS_READ_DVDRW_PLUS macro 375	BS_SCSI_ERROR_01 macro 380
BS_READ_HDDVD_R 375	BS_SCSI_ERROR_010 380
BS_READ_HDDVD_R macro 375	BS_SCSI_ERROR_010 macro 380
BS_READ_HDDVD_ROM 375	BS_SCSI_ERROR_011 381
BS_READ_HDDVD_ROM macro 375	BS_SCSI_ERROR_011 macro 381
BS_READ_HDDVD_RW 376	BS_SCSI_ERROR_012 381
BS_READ_HDDVD_RW macro 376	BS_SCSI_ERROR_012 macro 381
BS_READ_MOUNT_RAINER 376	BS_SCSI_ERROR_013 381
BS_READ_MOUNT_RAINER macro 376	BS_SCSI_ERROR_013 macro 381
BS_RM_RAW 376	BS_SCSI_ERROR_014 381
BS_RM_RAW macro 376	BS_SCSI_ERROR_014 macro 381
BS_RM_RAW_SUBCHANNEL 376	BS_SCSI_ERROR_015 382
BS_RM_RAW_SUBCHANNEL macro 376	BS_SCSI_ERROR_015 macro 382
BS_RM_USERDATA 377	BS_SCSI_ERROR_016 382
BS_RM_USERDATA macro 377	BS_SCSI_ERROR_016 macro 382
BS_RTF_AUDIO 377	BS_SCSI_ERROR_017 382
BS_RTF_AUDIO macro 377	BS_SCSI_ERROR_017 macro 382
BS_RTF_MODE1 377	BS_SCSI_ERROR_018 382
BS_RTF_MODE1 macro 377	BS_SCSI_ERROR_018 macro 382
BS_RTF_MODE2_FORM1 377	BS_SCSI_ERROR_019 383
BS_RTF_MODE2_FORM1 macro 377	BS_SCSI_ERROR_019 macro 383
BS_RTF_MODE2_FORM2 378	BS_SCSI_ERROR_02 383
BS_RTF_MODE2_FORM2 macro 378	BS_SCSI_ERROR_02 macro 383
BS_RTF_MODE2_FORMLESS 378	BS_SCSI_ERROR_020 383
BS_RTF_MODE2_FORMLESS macro 378	BS_SCSI_ERROR_020 macro 383
BS_SCSI_ERROR_001 378	BS_SCSI_ERROR_021 383
BS_SCSI_ERROR_001 macro 378	BS_SCSI_ERROR_021 macro 383
BS_SCSI_ERROR_002 378	BS_SCSI_ERROR_022 384
BS_SCSI_ERROR_002 macro 378	BS_SCSI_ERROR_022 macro 384
BS_SCSI_ERROR_004 379	BS_SCSI_ERROR_023 384
BS_SCSI_ERROR_004 macro 379	BS_SCSI_ERROR_023 macro 384
BS_SCSI_ERROR_005 379	BS_SCSI_ERROR_024 384
BS_SCSI_ERROR_005 macro 379	BS_SCSI_ERROR_024 macro 384
BS_SCSI_ERROR_006 379	BS_SCSI_ERROR_025 384
BS_SCSI_ERROR_006 macro 379	BS_SCSI_ERROR_025 macro 384
BS_SCSI_ERROR_007 379	BS_SCSI_ERROR_026 385
BS_SCSI_ERROR_007 macro 379	BS_SCSI_ERROR_026 macro 385
BS_SCSI_ERROR_008 380	BS_SCSI_ERROR_027 385
BS_SCSI_ERROR_008 macro 380	BS_SCSI_ERROR_027 macro 385
BS_SCSI_ERROR_009 380	BS_SCSI_ERROR_028 385

BS_SCSI_ERROR_028 macro 385	BS_SCSI_ERROR_049 macro 390
BS_SCSI_ERROR_029 385	BS_SCSI_ERROR_050 391
BS_SCSI_ERROR_029 macro 385	BS_SCSI_ERROR_050 macro 391
BS_SCSI_ERROR_030 386	BS_SCSI_ERROR_051 391
BS_SCSI_ERROR_030 macro 386	BS_SCSI_ERROR_051 macro 391
BS_SCSI_ERROR_031 386	BS_SCSI_ERROR_052 391
BS_SCSI_ERROR_031 macro 386	BS_SCSI_ERROR_052 macro 391
BS_SCSI_ERROR_032 386	BS_SCSI_ERROR_053 391
BS_SCSI_ERROR_032 macro 386	BS_SCSI_ERROR_053 macro 391
BS_SCSI_ERROR_033 386	BS_SCSI_ERROR_054 392
BS_SCSI_ERROR_033 macro 386	BS_SCSI_ERROR_054 macro 392
BS_SCSI_ERROR_034 387	BS_SCSI_ERROR_055 392
BS_SCSI_ERROR_034 macro 387	BS_SCSI_ERROR_055 macro 392
BS_SCSI_ERROR_035 387	BS_SCSI_ERROR_056 392
BS_SCSI_ERROR_036 387	BS_SCSI_ERROR_057 392
BS_SCSI_ERROR_036 macro 387	BS_SCSI_ERROR_057 macro 392
BS_SCSI_ERROR_037 387	BS_SCSI_ERROR_058 393
BS_SCSI_ERROR_037 macro 387	BS_SCSI_ERROR_058 macro 393
BS_SCSI_ERROR_038 388	BS_SCSI_ERROR_059 393
BS_SCSI_ERROR_038 macro 388	BS_SCSI_ERROR_059 macro 393
BS_SCSI_ERROR_039 388	BS_SCSI_ERROR_060 393
BS_SCSI_ERROR_039 macro 388	BS_SCSI_ERROR_060 macro 393
BS_SCSI_ERROR_040 388	BS_SCSI_ERROR_061 393
BS_SCSI_ERROR_040 macro 388	BS_SCSI_ERROR_061 macro 393
BS_SCSI_ERROR_041 388	BS_SCSI_ERROR_062 394
BS_SCSI_ERROR_041 macro 388	BS_SCSI_ERROR_062 macro 394
BS_SCSI_ERROR_042 389	BS_SCSI_ERROR_063 394
BS_SCSI_ERROR_042 macro 389	BS_SCSI_ERROR_063 macro 394
BS_SCSI_ERROR_043 389	BS_SCSI_ERROR_064 394
BS_SCSI_ERROR_043 macro 389	BS_SCSI_ERROR_064 macro 394
BS_SCSI_ERROR_044 389	BS_SCSI_ERROR_065 394
BS_SCSI_ERROR_044 macro 389	BS_SCSI_ERROR_065 macro 394
BS_SCSI_ERROR_045 389	BS_SCSI_ERROR_066 395
BS_SCSI_ERROR_045 macro 389	BS_SCSI_ERROR_066 macro 395
BS_SCSI_ERROR_046 390	BS_SCSI_ERROR_067 395
BS_SCSI_ERROR_046 macro 390	BS_SCSI_ERROR_067 macro 395
BS_SCSI_ERROR_047 390	BS_SCSI_ERROR_068 395
BS_SCSI_ERROR_047 macro 390	BS_SCSI_ERROR_068 macro 395
BS_SCSI_ERROR_048 390	BS_SCSI_ERROR_069 395
BS_SCSI_ERROR_048 macro 390	BS_SCSI_ERROR_069 macro 395
BS_SCSI_ERROR_049 390	BS_SCSI_ERROR_070 396

BS_SCSI_ERROR_070 macro 396	BS_SCSI_ERROR_091 macro 401
BS_SCSI_ERROR_071 396	BS_SCSI_ERROR_092 401
BS_SCSI_ERROR_071 macro 396	BS_SCSI_ERROR_092 macro 401
BS_SCSI_ERROR_072 396	BS_SCSI_ERROR_093 401
BS_SCSI_ERROR_072 macro 396	BS_SCSI_ERROR_093 macro 401
BS_SCSI_ERROR_073 396	BS_SCSI_ERROR_094 402
BS_SCSI_ERROR_074 397	BS_SCSI_ERROR_095 402
BS_SCSI_ERROR_074 macro 397	BS_SCSI_ERROR_095 macro 402
BS_SCSI_ERROR_075 397	BS_SCSI_ERROR_096 402
BS_SCSI_ERROR_075 macro 397	BS_SCSI_ERROR_096 macro 402
BS_SCSI_ERROR_076 397	BS_SCSI_ERROR_097 402
BS_SCSI_ERROR_076 macro 397	BS_SCSI_ERROR_097 macro 402
BS_SCSI_ERROR_077 397	BS_SCSI_ERROR_098 403
BS_SCSI_ERROR_077 macro 397	BS_SCSI_ERROR_098 macro 403
BS_SCSI_ERROR_078 398	BS_SCSI_ERROR_099 403
BS_SCSI_ERROR_078 macro 398	BS_SCSI_ERROR_099 macro 403
BS_SCSI_ERROR_079 398	BS_SCSI_ERROR_100 403
BS_SCSI_ERROR_079 macro 398	BS_SCSI_ERROR_100 macro 403
BS_SCSI_ERROR_080 398	BS_SCSI_ERROR_101 403
BS_SCSI_ERROR_080 macro 398	BS_SCSI_ERROR_101 macro 403
BS_SCSI_ERROR_081 398	BS_SCSI_ERROR_102 404
BS_SCSI_ERROR_081 macro 398	BS_SCSI_ERROR_102 macro 404
BS_SCSI_ERROR_082 399	BS_SCSI_ERROR_103 404
BS_SCSI_ERROR_082 macro 399	BS_SCSI_ERROR_103 macro 404
BS_SCSI_ERROR_083 399	BS_SCSI_ERROR_104 404
BS_SCSI_ERROR_083 macro 399	BS_SCSI_ERROR_104 macro 404
BS_SCSI_ERROR_084 399	BS_SCSI_ERROR_105 404
BS_SCSI_ERROR_084 macro 399	BS_SCSI_ERROR_105 macro 404
BS_SCSI_ERROR_085 399	BS_SCSI_ERROR_106 405
BS_SCSI_ERROR_085 macro 399	BS_SCSI_ERROR_106 macro 405
BS_SCSI_ERROR_086 400	BS_SCSI_ERROR_107 405
BS_SCSI_ERROR_086 macro 400	BS_SCSI_ERROR_107 macro 405
BS_SCSI_ERROR_087 400	BS_SCSI_ERROR_108 405
BS_SCSI_ERROR_087 macro 400	BS_SCSI_ERROR_108 macro 405
BS_SCSI_ERROR_088 400	BS_SCSI_ERROR_109 405
BS_SCSI_ERROR_088 macro 400	BS_SCSI_ERROR_109 macro 405
BS_SCSI_ERROR_089 400	BS_SCSI_ERROR_110 406
BS_SCSI_ERROR_089 macro 400	BS_SCSI_ERROR_110 macro 406
BS_SCSI_ERROR_090 401	BS_SCSI_ERROR_111 406
BS_SCSI_ERROR_090 macro 401	BS_SCSI_ERROR_111 macro 406
BS_SCSI_ERROR_091 401	BS_SCSI_ERROR_112 406

BS\_SCSI\_ERROR\_112 macro 406  
BS\_SCSI\_ERROR\_113 406  
BS\_SCSI\_ERROR\_113 macro 406  
BS\_SCSI\_ERROR\_114 407  
BS\_SCSI\_ERROR\_114 macro 407  
BS\_SCSI\_ERROR\_115 407  
BS\_SCSI\_ERROR\_115 macro 407  
BS\_SCSI\_ERROR\_116 407  
BS\_SCSI\_ERROR\_116 macro 407  
BS\_SCSI\_ERROR\_117 407  
BS\_SCSI\_ERROR\_117 macro 407  
BS\_SCSI\_ERROR\_118 408  
BS\_SCSI\_ERROR\_118 macro 408  
BS\_SCSI\_ERROR\_119 408  
BS\_SCSI\_ERROR\_119 macro 408  
BS\_SCSI\_ERROR\_ALLOC\_01 408  
BS\_SCSI\_ERROR\_ALLOC\_01 macro 408  
BS\_SCSI\_ERROR\_ALLOC\_02 408  
BS\_SCSI\_ERROR\_ALLOC\_02 macro 408  
BS\_SCSI\_ERROR\_ASPI\_01 409  
BS\_SCSI\_ERROR\_ASPI\_01 macro 409  
BS\_SCSI\_ERROR\_ASPI\_02 409  
BS\_SCSI\_ERROR\_ASPI\_02 macro 409  
BS\_SCSI\_ERROR\_ASPI\_03 409  
BS\_SCSI\_ERROR\_ASPI\_03 macro 409  
BS\_SCSI\_ERROR\_ASPI\_04 409  
BS\_SCSI\_ERROR\_ASPI\_04 macro 409  
BS\_SCSI\_ERROR\_ASPI\_05 410  
BS\_SCSI\_ERROR\_ASPI\_05 macro 410  
BS\_SCSI\_ERROR\_ASPI\_06 410  
BS\_SCSI\_ERROR\_ASPI\_06 macro 410  
BS\_SCSI\_ERROR\_ASPI\_07 410  
BS\_SCSI\_ERROR\_ASPI\_07 macro 410  
BS\_SCSI\_ERROR\_ASPI\_08 410  
BS\_SCSI\_ERROR\_ASPI\_08 macro 410  
BS\_SCSI\_ERROR\_ASPI\_09 411  
BS\_SCSI\_ERROR\_ASPI\_09 macro 411  
BS\_SCSI\_ERROR\_ASPI\_10 411  
BS\_SCSI\_ERROR\_ASPI\_10 macro 411  
BS\_SCSI\_ERROR\_ATT\_01 411  
BS\_SCSI\_ERROR\_ATT\_01 macro 411  
BS\_SCSI\_ERROR\_ATT\_02 411  
BS\_SCSI\_ERROR\_ATT\_02 macro 411  
BS\_SCSI\_ERROR\_ATT\_03 412  
BS\_SCSI\_ERROR\_ATT\_03 macro 412  
BS\_SCSI\_ERROR\_ATT\_04 412  
BS\_SCSI\_ERROR\_ATT\_04 macro 412  
BS\_SCSI\_ERROR\_ATT\_05 412  
BS\_SCSI\_ERROR\_ATT\_05 macro 412  
BS\_SCSI\_ERROR\_AUDIO\_01 412  
BS\_SCSI\_ERROR\_AUDIO\_01 macro 412  
BS\_SCSI\_ERROR\_AUDIO\_02 413  
BS\_SCSI\_ERROR\_AUDIO\_02 macro 413  
BS\_SCSI\_ERROR\_AUDIO\_03 413  
BS\_SCSI\_ERROR\_AUDIO\_03 macro 413  
BS\_SCSI\_ERROR\_AUDIO\_04 413  
BS\_SCSI\_ERROR\_AUDIO\_04 macro 413  
BS\_SCSI\_ERROR\_AUDIO\_05 413  
BS\_SCSI\_ERROR\_AUDIO\_05 macro 413  
BS\_SCSI\_ERROR\_CDB\_01 414  
BS\_SCSI\_ERROR\_CDB\_01 macro 414  
BS\_SCSI\_ERROR\_CDB\_02 414  
BS\_SCSI\_ERROR\_CDB\_02 macro 414  
BS\_SCSI\_ERROR\_CIRC\_01 414  
BS\_SCSI\_ERROR\_CIRC\_01 macro 414  
BS\_SCSI\_ERROR\_COMMAND\_02 414  
BS\_SCSI\_ERROR\_COMMAND\_02 macro 414  
BS\_SCSI\_ERROR\_COMMAND\_03 415  
BS\_SCSI\_ERROR\_COMMAND\_03 macro 415  
BS\_SCSI\_ERROR\_COMMAND\_04 415  
BS\_SCSI\_ERROR\_COMMAND\_04 macro 415  
BS\_SCSI\_ERROR\_COMMAND\_05 415  
BS\_SCSI\_ERROR\_COMMAND\_05 macro 415  
BS\_SCSI\_ERROR\_CRC\_01 415  
BS\_SCSI\_ERROR\_CRC\_01 macro 415  
BS\_SCSI\_ERROR\_DCSS\_01 416  
BS\_SCSI\_ERROR\_DCSS\_01 macro 416  
BS\_SCSI\_ERROR\_DCSS\_02 416  
BS\_SCSI\_ERROR\_DCSS\_02 macro 416  
BS\_SCSI\_ERROR\_DCSS\_03 416  
BS\_SCSI\_ERROR\_DCSS\_03 macro 416  
BS\_SCSI\_ERROR\_DCSS\_04 416  
BS\_SCSI\_ERROR\_DCSS\_04 macro 416  
BS\_SCSI\_ERROR\_DCSS\_05 417

BS\_SCSI\_ERROR\_DCSS\_05 macro 417  
BS\_SCSI\_ERROR\_DCSS\_06 417  
BS\_SCSI\_ERROR\_DCSS\_06 macro 417  
BS\_SCSI\_ERROR\_DCSS\_07 417  
BS\_SCSI\_ERROR\_DCSS\_07 macro 417  
BS\_SCSI\_ERROR\_DCSS\_08 417  
BS\_SCSI\_ERROR\_DCSS\_08 macro 417  
BS\_SCSI\_ERROR\_DECOM\_01 418  
BS\_SCSI\_ERROR\_DECOM\_01 macro 418  
BS\_SCSI\_ERROR\_DISK\_01 418  
BS\_SCSI\_ERROR\_DISK\_01 macro 418  
BS\_SCSI\_ERROR\_DISK\_02 418  
BS\_SCSI\_ERROR\_DISK\_02 macro 418  
BS\_SCSI\_ERROR\_DISK\_03 418  
BS\_SCSI\_ERROR\_DISK\_03 macro 418  
BS\_SCSI\_ERROR\_DISK\_04 419  
BS\_SCSI\_ERROR\_DISK\_04 macro 419  
BS\_SCSI\_ERROR\_DISK\_05 419  
BS\_SCSI\_ERROR\_DISK\_05 macro 419  
BS\_SCSI\_ERROR\_DISK\_06 419  
BS\_SCSI\_ERROR\_DISK\_06 macro 419  
BS\_SCSI\_ERROR\_DISK\_07 419  
BS\_SCSI\_ERROR\_DISK\_07 macro 419  
BS\_SCSI\_ERROR\_DISK\_08 420  
BS\_SCSI\_ERROR\_DISK\_08 macro 420  
BS\_SCSI\_ERROR\_DISK\_09 420  
BS\_SCSI\_ERROR\_DISK\_09 macro 420  
BS\_SCSI\_ERROR\_DISK\_10 420  
BS\_SCSI\_ERROR\_DISK\_10 macro 420  
BS\_SCSI\_ERROR\_DISK\_11 420  
BS\_SCSI\_ERROR\_DISK\_11 macro 420  
BS\_SCSI\_ERROR\_DISK\_12 421  
BS\_SCSI\_ERROR\_DISK\_12 macro 421  
BS\_SCSI\_ERROR\_DISK\_13 421  
BS\_SCSI\_ERROR\_DISK\_13 macro 421  
BS\_SCSI\_ERROR\_DISK\_14 421  
BS\_SCSI\_ERROR\_DISK\_14 macro 421  
BS\_SCSI\_ERROR\_DISK\_15 421  
BS\_SCSI\_ERROR\_DISK\_15 macro 421  
BS\_SCSI\_ERROR\_DISK\_16 422  
BS\_SCSI\_ERROR\_DISK\_16 macro 422  
BS\_SCSI\_ERROR\_DISK\_17 422  
BS\_SCSI\_ERROR\_DISK\_17 macro 422  
BS\_SCSI\_ERROR\_DISK\_18 422  
BS\_SCSI\_ERROR\_DISK\_18 macro 422  
BS\_SCSI\_ERROR\_DISK\_19 422  
BS\_SCSI\_ERROR\_DISK\_19 macro 422  
BS\_SCSI\_ERROR\_DISK\_20 423  
BS\_SCSI\_ERROR\_DISK\_20 macro 423  
BS\_SCSI\_ERROR\_DISK\_21 423  
BS\_SCSI\_ERROR\_DISK\_21 macro 423  
BS\_SCSI\_ERROR\_DISK\_22 423  
BS\_SCSI\_ERROR\_DISK\_22 macro 423  
BS\_SCSI\_ERROR\_DISK\_23 423  
BS\_SCSI\_ERROR\_DISK\_23 macro 423  
BS\_SCSI\_ERROR\_DISK\_24 424  
BS\_SCSI\_ERROR\_DISK\_24 macro 424  
BS\_SCSI\_ERROR\_DISK\_25 424  
BS\_SCSI\_ERROR\_DISK\_25 macro 424  
BS\_SCSI\_ERROR\_DISK\_26 424  
BS\_SCSI\_ERROR\_DISK\_26 macro 424  
BS\_SCSI\_ERROR\_DISK\_27 424  
BS\_SCSI\_ERROR\_DISK\_27 macro 424  
BS\_SCSI\_ERROR\_DISK\_28 425  
BS\_SCSI\_ERROR\_DISK\_28 macro 425  
BS\_SCSI\_ERROR\_DISK\_29 425  
BS\_SCSI\_ERROR\_DISK\_29 macro 425  
BS\_SCSI\_ERROR\_DISK\_30 425  
BS\_SCSI\_ERROR\_DISK\_30 macro 425  
BS\_SCSI\_ERROR\_DISK\_31 425  
BS\_SCSI\_ERROR\_DISK\_31 macro 425  
BS\_SCSI\_ERROR\_DISK\_32 426  
BS\_SCSI\_ERROR\_DISK\_32 macro 426  
BS\_SCSI\_ERROR\_DISK\_33 426  
BS\_SCSI\_ERROR\_DISK\_33 macro 426  
BS\_SCSI\_ERROR\_DISK\_34 426  
BS\_SCSI\_ERROR\_DISK\_34 macro 426  
BS\_SCSI\_ERROR\_DISK\_35 426  
BS\_SCSI\_ERROR\_DISK\_35 macro 426  
BS\_SCSI\_ERROR\_DISK\_36 427  
BS\_SCSI\_ERROR\_DISK\_36 macro 427  
BS\_SCSI\_ERROR\_DISK\_37 427  
BS\_SCSI\_ERROR\_DISK\_37 macro 427  
BS\_SCSI\_ERROR\_DRIVE\_01 427

BS\_SCSI\_ERROR\_DRIVE\_01 macro 427  
BS\_SCSI\_ERROR\_DRIVE\_02 427  
BS\_SCSI\_ERROR\_DRIVE\_02 macro 427  
BS\_SCSI\_ERROR\_EXT\_01 428  
BS\_SCSI\_ERROR\_EXT\_01 macro 428  
BS\_SCSI\_ERROR\_EXT\_02 428  
BS\_SCSI\_ERROR\_EXT\_02 macro 428  
BS\_SCSI\_ERROR\_LOG\_01 428  
BS\_SCSI\_ERROR\_LOG\_01 macro 428  
BS\_SCSI\_ERROR\_LOG\_02 428  
BS\_SCSI\_ERROR\_LOG\_02 macro 428  
BS\_SCSI\_ERROR\_LOG\_03 429  
BS\_SCSI\_ERROR\_LOG\_03 macro 429  
BS\_SCSI\_ERROR\_LOG\_04 429  
BS\_SCSI\_ERROR\_LOG\_04 macro 429  
BS\_SCSI\_ERROR\_MECH\_01 429  
BS\_SCSI\_ERROR\_MECH\_01 macro 429  
BS\_SCSI\_ERROR\_MECH\_02 429  
BS\_SCSI\_ERROR\_MECH\_02 macro 429  
BS\_SCSI\_ERROR\_MECH\_03 430  
BS\_SCSI\_ERROR\_MECH\_03 macro 430  
BS\_SCSI\_ERROR\_PARAM\_01 430  
BS\_SCSI\_ERROR\_PARAM\_01 macro 430  
BS\_SCSI\_ERROR\_PARAM\_02 430  
BS\_SCSI\_ERROR\_PARAM\_02 macro 430  
BS\_SCSI\_ERROR\_PARAM\_03 430  
BS\_SCSI\_ERROR\_PARAM\_03 macro 430  
BS\_SCSI\_ERROR\_PARAM\_04 431  
BS\_SCSI\_ERROR\_PARAM\_04 macro 431  
BS\_SCSI\_ERROR\_PARAM\_05 431  
BS\_SCSI\_ERROR\_PARAM\_05 macro 431  
BS\_SCSI\_ERROR\_PARAM\_06 431  
BS\_SCSI\_ERROR\_PARAM\_06 macro 431  
BS\_SCSI\_ERROR\_PARAM\_07 431  
BS\_SCSI\_ERROR\_PARAM\_07 macro 431  
BS\_SCSI\_ERROR\_PARAM\_08 432  
BS\_SCSI\_ERROR\_PARAM\_08 macro 432  
BS\_SCSI\_ERROR\_PARAM\_09 432  
BS\_SCSI\_ERROR\_PARAM\_09 macro 432  
BS\_SCSI\_ERROR\_PARAM\_10 432  
BS\_SCSI\_ERROR\_PARAM\_10 macro 432  
BS\_SCSI\_ERROR\_READ\_01 432  
BS\_SCSI\_ERROR\_READ\_01 macro 432  
BS\_SCSI\_ERROR\_READ\_02 433  
BS\_SCSI\_ERROR\_READ\_02 macro 433  
BS\_SCSI\_ERROR\_RECOVER\_01 433  
BS\_SCSI\_ERROR\_RECOVER\_01 macro 433  
BS\_SCSI\_ERROR\_RECOVER\_02 433  
BS\_SCSI\_ERROR\_RECOVER\_02 macro 433  
BS\_SCSI\_ERROR\_RECOVER\_03 433  
BS\_SCSI\_ERROR\_RECOVER\_03 macro 433  
BS\_SCSI\_ERROR\_RECOVER\_04 434  
BS\_SCSI\_ERROR\_RECOVER\_04 macro 434  
BS\_SCSI\_ERROR\_RECOVER\_05 434  
BS\_SCSI\_ERROR\_RECOVER\_05 macro 434  
BS\_SCSI\_ERROR\_RECOVER\_06 434  
BS\_SCSI\_ERROR\_RECOVER\_06 macro 434  
BS\_SCSI\_ERROR\_RECOVER\_07 434  
BS\_SCSI\_ERROR\_RECOVER\_07 macro 434  
BS\_SCSI\_ERROR\_RECOVER\_08 435  
BS\_SCSI\_ERROR\_RECOVER\_08 macro 435  
BS\_SCSI\_ERROR\_RECOVER\_09 435  
BS\_SCSI\_ERROR\_RECOVER\_09 macro 435  
BS\_SCSI\_ERROR\_RECOVER\_10 435  
BS\_SCSI\_ERROR\_RECOVER\_10 macro 435  
BS\_SCSI\_ERROR\_RECOVER\_11 435  
BS\_SCSI\_ERROR\_RECOVER\_11 macro 435  
BS\_SCSI\_ERROR\_RECOVER\_12 436  
BS\_SCSI\_ERROR\_RECOVER\_12 macro 436  
BS\_SCSI\_ERROR\_RECOVER\_13 436  
BS\_SCSI\_ERROR\_RECOVER\_13 macro 436  
BS\_SCSI\_ERROR\_RECOVER\_14 436  
BS\_SCSI\_ERROR\_RECOVER\_14 macro 436  
BS\_SCSI\_ERROR\_RECOVER\_15 436  
BS\_SCSI\_ERROR\_RECOVER\_15 macro 436  
BS\_SCSI\_ERROR\_RECOVER\_16 437  
BS\_SCSI\_ERROR\_RECOVER\_16 macro 437  
BS\_SCSI\_ERROR\_RECOVER\_17 437  
BS\_SCSI\_ERROR\_RECOVER\_17 macro 437  
BS\_SCSI\_ERROR\_SEGM\_01 437  
BS\_SCSI\_ERROR\_SEGM\_01 macro 437  
BS\_SCSI\_ERROR\_SEGM\_02 437  
BS\_SCSI\_ERROR\_SEGM\_02 macro 437  
BS\_SCSI\_ERROR\_SEGM\_03 438

BS\_SCSI\_ERROR\_SEGM\_03 macro 438  
BS\_SCSI\_ERROR\_SEGM\_04 438  
BS\_SCSI\_ERROR\_SEGM\_04 macro 438  
BS\_SCSI\_ERROR\_SESSION\_01 438  
BS\_SCSI\_ERROR\_SESSION\_01 macro 438  
BS\_SCSI\_ERROR\_SESSION\_02 438  
BS\_SCSI\_ERROR\_SESSION\_02 macro 438  
BS\_SCSI\_ERROR\_SESSION\_03 439  
BS\_SCSI\_ERROR\_SESSION\_03 macro 439  
BS\_SCSI\_ERROR\_SESSION\_04 439  
BS\_SCSI\_ERROR\_SESSION\_04 macro 439  
BS\_SCSI\_ERROR\_TARGET\_01 439  
BS\_SCSI\_ERROR\_TARGET\_01 macro 439  
BS\_SCSI\_ERROR\_TARGET\_02 439  
BS\_SCSI\_ERROR\_TARGET\_02 macro 439  
BS\_SCSI\_ERROR\_TARGET\_03 440  
BS\_SCSI\_ERROR\_TARGET\_03 macro 440  
BS\_SCSI\_ERROR\_TARGET\_04 440  
BS\_SCSI\_ERROR\_TARGET\_04 macro 440  
BS\_SCSI\_ERROR\_UNIT\_01 440  
BS\_SCSI\_ERROR\_UNIT\_01 macro 440  
BS\_SCSI\_ERROR\_UNIT\_02 440  
BS\_SCSI\_ERROR\_UNIT\_02 macro 440  
BS\_SCSI\_ERROR\_UNIT\_03 441  
BS\_SCSI\_ERROR\_UNIT\_03 macro 441  
BS\_SCSI\_ERROR\_UNIT\_04 441  
BS\_SCSI\_ERROR\_UNIT\_04 macro 441  
BS\_SCSI\_ERROR\_UNIT\_05 441  
BS\_SCSI\_ERROR\_UNIT\_05 macro 441  
BS\_SCSI\_ERROR\_UNIT\_06 441  
BS\_SCSI\_ERROR\_UNIT\_06 macro 441  
BS\_SCSI\_ERROR\_UNIT\_07 442  
BS\_SCSI\_ERROR\_UNIT\_07 macro 442  
BS\_SCSI\_ERROR\_UNIT\_08 442  
BS\_SCSI\_ERROR\_UNIT\_08 macro 442  
BS\_SCSI\_ERROR\_UNIT\_09 442  
BS\_SCSI\_ERROR\_UNIT\_09 macro 442  
BS\_SCSI\_ERROR\_UNIT\_10 442  
BS\_SCSI\_ERROR\_UNIT\_10 macro 442  
BS\_SCSI\_ERROR\_UNIT\_11 443  
BS\_SCSI\_ERROR\_UNIT\_11 macro 443  
BS\_SCSI\_ERROR\_UNIT\_12 443  
BS\_SCSI\_ERROR\_UNIT\_12 macro 443  
BS\_SCSI\_ERROR\_UNIT\_13 443  
BS\_SCSI\_ERROR\_UNIT\_13 macro 443  
BS\_SCSI\_ERROR\_UNIT\_14 443  
BS\_SCSI\_ERROR\_UNIT\_14 macro 443  
BS\_SCSI\_ERROR\_UNIT\_16 444  
BS\_SCSI\_ERROR\_UNIT\_16 macro 444  
BS\_SCSI\_ERROR\_UNIT\_17 444  
BS\_SCSI\_ERROR\_UNIT\_17 macro 444  
BS\_SCSI\_ERROR\_UNIT\_18 444  
BS\_SCSI\_ERROR\_UNIT\_18 macro 444  
BS\_SCSI\_ERROR\_UNIT\_19 444  
BS\_SCSI\_ERROR\_UNIT\_19 macro 444  
BS\_SCSI\_ERROR\_UNIT\_20 445  
BS\_SCSI\_ERROR\_UNIT\_20 macro 445  
BS\_SCSI\_ERROR\_UNIT\_21 445  
BS\_SCSI\_ERROR\_UNIT\_21 macro 445  
BS\_SCSI\_ERROR\_UNIT\_22 445  
BS\_SCSI\_ERROR\_UNIT\_22 macro 445  
BS\_SCSI\_ERROR\_UNIT\_23 445  
BS\_SCSI\_ERROR\_UNIT\_23 macro 445  
BS\_SCSI\_ERROR\_VOL\_01 446  
BS\_SCSI\_ERROR\_VOL\_01 macro 446  
BS\_SCSI\_ERROR\_VOL\_02 446  
BS\_SCSI\_ERROR\_VOL\_02 macro 446  
BS\_SCSI\_ERROR\_VOL\_03 446  
BS\_SCSI\_ERROR\_VOL\_03 macro 446  
BS\_SCSI\_ERROR\_VOL\_04 446  
BS\_SCSI\_ERROR\_VOL\_04 macro 446  
BS\_SCSI\_ERROR\_WRITE\_01 447  
BS\_SCSI\_ERROR\_WRITE\_01 macro 447  
BS\_SCSI\_ERROR\_WRITE\_02 447  
BS\_SCSI\_ERROR\_WRITE\_02 macro 447  
BS\_SCSI\_ERROR\_WRITE\_03 447  
BS\_SCSI\_ERROR\_WRITE\_03 macro 447  
BS\_SCSI\_ERROR\_WRITE\_04 447  
BS\_SCSI\_ERROR\_WRITE\_04 macro 447  
BS\_SCSI\_ERROR\_WRITE\_05 448  
BS\_SCSI\_ERROR\_WRITE\_05 macro 448  
BS\_SCSI\_ERROR\_WRITE\_06 448  
BS\_SCSI\_ERROR\_WRITE\_06 macro 448  
BS\_SCSI\_ERROR\_WRITE\_07 448

BS\_SCSI\_ERROR\_WRITE\_07 macro 448  
BS\_SCSI\_ERROR\_WRITE\_08 448  
BS\_SCSI\_ERROR\_WRITE\_08 macro 448  
BS\_SCSI\_ERROR\_WRITE\_09 449  
BS\_SCSI\_ERROR\_WRITE\_09 macro 449  
BS\_SCSI\_ERROR\_WRITE\_10 449  
BS\_SCSI\_ERROR\_WRITE\_10 macro 449  
BS\_SCSI\_ERROR\_WRITE\_11 449  
BS\_SCSI\_ERROR\_WRITE\_11 macro 449  
BS\_SCSI\_ERROR\_WRITE\_12 449  
BS\_SCSI\_ERROR\_WRITE\_12 macro 449  
BS\_SDK\_COMPENC\_BOTH 450  
BS\_SDK\_COMPENC\_BOTH macro 450  
BS\_SDK\_COMPENC\_COMPRESSED 450  
BS\_SDK\_COMPENC\_COMPRESSED macro 450  
BS\_SDK\_COMPENC\_ENCRYPTED 450  
BS\_SDK\_COMPENC\_ENCRYPTED macro 450  
BS\_SDK\_CUE\_ERROR\_COMMAND\_01 450  
BS\_SDK\_CUE\_ERROR\_COMMAND\_01 macro 450  
BS\_SDK\_CUE\_ERROR\_COMMAND\_06 451  
BS\_SDK\_CUE\_ERROR\_COMMAND\_06 macro 451  
BS\_SDK\_CUE\_ERROR\_FIELD 451  
BS\_SDK\_CUE\_ERROR\_FIELD macro 451  
BS\_SDK\_CUE\_ERROR\_FILE 451  
BS\_SDK\_CUE\_ERROR\_FILE macro 451  
BS\_SDK\_CUE\_ERROR\_SENDING\_CUE 451  
BS\_SDK\_CUE\_ERROR\_SENDING\_CUE macro 451  
BS\_SDK\_CUE\_ERROR\_UEOI 452  
BS\_SDK\_CUE\_ERROR\_UEOI macro 452  
BS\_SDK\_ERROR\_ABORTED 452  
BS\_SDK\_ERROR\_ABORTED macro 452  
BS\_SDK\_ERROR\_ABORTEDCOMMAND 452  
BS\_SDK\_ERROR\_ABORTEDCOMMAND macro 452  
BS\_SDK\_ERROR\_BAD\_CAPABILITY\_NAME 452  
BS\_SDK\_ERROR\_BAD\_CAPABILITY\_NAME macro 452  
BS\_SDK\_ERROR\_BAD\_REQUEST 453  
BS\_SDK\_ERROR\_BAD\_REQUEST macro 453  
BS\_SDK\_ERROR\_BIN\_FILE\_NOT\_FOUND 453  
BS\_SDK\_ERROR\_BIN\_FILE\_NOT\_FOUND macro 453  
BS\_SDK\_ERROR\_BUFFERALIGNMENT 453  
BS\_SDK\_ERROR\_BUFFERALIGNMENT macro 453  
BS\_SDK\_ERROR\_BUFFERTOOBIG 453  
BS\_SDK\_ERROR\_BUFFERTOOBIG macro 453  
BS\_SDK\_ERROR\_BURN\_IN\_PROGRESS 454  
BS\_SDK\_ERROR\_BURN\_IN\_PROGRESS macro 454  
BS\_SDK\_ERROR\_CDTEXT\_NOT\_FOUND 454  
BS\_SDK\_ERROR\_CDTEXT\_NOT\_FOUND macro 454  
BS\_SDK\_ERROR\_CHECKCONDITION 454  
BS\_SDK\_ERROR\_CHECKCONDITION macro 454  
BS\_SDK\_ERROR\_COMPENC\_GENERALERROR 454  
BS\_SDK\_ERROR\_COMPENC\_GENERALERROR macro 454  
BS\_SDK\_ERROR\_COMPENC\_READ10\_IDENT 455  
BS\_SDK\_ERROR\_COMPENC\_READ10\_IDENT macro 455  
BS\_SDK\_ERROR\_COMPRESSION\_CONFLICT 455  
BS\_SDK\_ERROR\_COMPRESSION\_CONFLICT macro 455  
BS\_SDK\_ERROR\_COMPRESSION\_NOINDXTABLE 455  
BS\_SDK\_ERROR\_COMPRESSION\_NOINDXTABLE macro 455  
BS\_SDK\_ERROR\_COMPRESSION\_NOSIGNATURE 455  
BS\_SDK\_ERROR\_COMPRESSION\_NOSIGNATURE macro 455  
BS\_SDK\_ERROR\_COMPRESSION\_READ10 456  
BS\_SDK\_ERROR\_COMPRESSION\_READ10 macro 456  
BS\_SDK\_ERROR\_COMPRESSION\_READMEMF 456  
BS\_SDK\_ERROR\_COMPRESSION\_READMEMF macro 456  
BS\_SDK\_ERROR\_COMPRESSION\_SEEKMEMFILE 456  
BS\_SDK\_ERROR\_COMPRESSION\_SEEKMEMFILE macro 456  
BS\_SDK\_ERROR\_COPYABORTED 457  
BS\_SDK\_ERROR\_COPYABORTED macro 457  
BS\_SDK\_ERROR\_CORRUPT\_OR\_INVALID\_CUE\_FILE 457  
BS\_SDK\_ERROR\_CORRUPT\_OR\_INVALID\_CUE\_FILE macro 457  
BS\_SDK\_ERROR\_CREATEFILE 457  
BS\_SDK\_ERROR\_CREATEFILE macro 457  
BS\_SDK\_ERROR\_DATAOVERRUN 457  
BS\_SDK\_ERROR\_DATAOVERRUN macro 457  
BS\_SDK\_ERROR\_DATAPROTECT 458  
BS\_SDK\_ERROR\_DATAPROTECT macro 458  
BS\_SDK\_ERROR\_DATARECOVERERROR 458  
BS\_SDK\_ERROR\_DATARECOVERERROR macro 458  
BS\_SDK\_ERROR\_DEVICE\_LOCKED 458  
BS\_SDK\_ERROR\_DEVICE\_LOCKED macro 458  
BS\_SDK\_ERROR\_EMPTY\_PASSWORD 458  
BS\_SDK\_ERROR\_EMPTY\_PASSWORD macro 458

BS_SDK_ERROR_ENCLIB_NOT_FOUND 459	BS_SDK_ERROR_INVALID_MCN 464
BS_SDK_ERROR_ENCLIB_NOT_FOUND macro 459	BS_SDK_ERROR_INVALID_MCN macro 464
BS_SDK_ERROR_ENCRYPTION_CONFLICT 459	BS_SDK_ERROR_INVALID_PATH 464
BS_SDK_ERROR_ENCRYPTION_CONFLICT macro 459	BS_SDK_ERROR_INVALID_PATH macro 464
BS_SDK_ERROR_ERASECHECK 459	BS_SDK_ERROR_INVALID_SESSION_NUMBER 464
BS_SDK_ERROR_ERASECHECK macro 459	BS_SDK_ERROR_INVALID_SESSION_NUMBER macro 464
BS_SDK_ERROR_ERRINVALIDFILENAME 459	BS_SDK_ERROR_INVALID_SRC_PATH 465
BS_SDK_ERROR_ERRINVALIDFILENAME macro 459	BS_SDK_ERROR_INVALID_SRC_PATH macro 465
BS_SDK_ERROR_FILE_EXISTS 460	BS_SDK_ERROR_INVALID_UDF_VERSION 465
BS_SDK_ERROR_FILE_EXISTS macro 460	BS_SDK_ERROR_INVALID_UDF_VERSION macro 465
BS_SDK_ERROR_FILE_OPEN 460	BS_SDK_ERROR_INVALIDSRB 465
BS_SDK_ERROR_FILE_OPEN macro 460	BS_SDK_ERROR_INVALIDSRB macro 465
BS_SDK_ERROR_FILEINUSE 460	BS_SDK_ERROR_ISOIMAGENOTFOUND 465
BS_SDK_ERROR_FILEINUSE macro 460	BS_SDK_ERROR_ISOIMAGENOTFOUND macro 465
BS_SDK_ERROR_FILEMARK 460	BS_SDK_ERROR_MAXDIRS 466
BS_SDK_ERROR_FILEMARK macro 460	BS_SDK_ERROR_MAXDIRS macro 466
BS_SDK_ERROR_GENERAL 461	BS_SDK_ERROR_MAXFILES 466
BS_SDK_ERROR_GENERAL macro 461	BS_SDK_ERROR_MAXFILES macro 466
BS_SDK_ERROR_HARDWAREERROR 461	BS_SDK_ERROR_MESSAGEJECT 466
BS_SDK_ERROR_HARDWAREERROR macro 461	BS_SDK_ERROR_MESSAGEJECT macro 466
BS_SDK_ERROR_ILLEGALLENGTH 461	BS_SDK_ERROR_MISCOMPARE 466
BS_SDK_ERROR_ILLEGALLENGTH macro 461	BS_SDK_ERROR_MISCOMPARE macro 466
BS_SDK_ERROR_IMPORTSESSION 461	BS_SDK_ERROR_MORE_SPACE_NEEDED 467
BS_SDK_ERROR_IMPORTSESSION macro 461	BS_SDK_ERROR_MORE_SPACE_NEEDED macro 467
BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE 462	BS_SDK_ERROR_MP3LIB_NOT_FOUND 467
BS_SDK_ERROR_INCOMPATIBLE_FS_TYPE macro 462	BS_SDK_ERROR_MP3LIB_NOT_FOUND macro 467
BS_SDK_ERROR_INCORRECTLENGTH 462	BS_SDK_ERROR_NETTAGS_CONNECT 467
BS_SDK_ERROR_INCORRECTLENGTH macro 462	BS_SDK_ERROR_NETTAGS_CONNECT macro 467
BS_SDK_ERROR_INVALID_DEST_PATH 462	BS_SDK_ERROR_NETTAGS_DISK 467
BS_SDK_ERROR_INVALID_DEST_PATH macro 462	BS_SDK_ERROR_NETTAGS_DISK macro 467
BS_SDK_ERROR_INVALID_DIR_INDEX 462	BS_SDK_ERROR_NETTAGS_INTERNAL 468
BS_SDK_ERROR_INVALID_DIR_INDEX macro 462	BS_SDK_ERROR_NETTAGS_INTERNAL macro 468
BS_SDK_ERROR_INVALID_FILE_FORMAT 463	BS_SDK_ERROR_NETTAGS_NOMATCH 468
BS_SDK_ERROR_INVALID_FILE_FORMAT macro 463	BS_SDK_ERROR_NETTAGS_NOMATCH macro 468
BS_SDK_ERROR_INVALID_FILE_NAME 463	BS_SDK_ERROR_NETTAGS_SERVER 468
BS_SDK_ERROR_INVALID_FILE_NAME macro 463	BS_SDK_ERROR_NETTAGS_SERVER macro 468
BS_SDK_ERROR_INVALID_HANDLE 463	BS_SDK_ERROR_NEXTADDRESS 468
BS_SDK_ERROR_INVALID_HANDLE macro 463	BS_SDK_ERROR_NEXTADDRESS macro 468
BS_SDK_ERROR_INVALID_INDEX 463	BS_SDK_ERROR_NO 469
BS_SDK_ERROR_INVALID_INDEX macro 463	BS_SDK_ERROR_NO macro 469
BS_SDK_ERROR_INVALID_ISRC 464	BS_SDK_ERROR_NOT_ALLOWED 469
BS_SDK_ERROR_INVALID_ISRC macro 464	BS_SDK_ERROR_NOT_ALLOWED macro 469

BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER 469	BS_SDK_ERROR_UNKNOWN_TEXTID 474
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_BURNER macro 469	BS_SDK_ERROR_UNKNOWN_TEXTID macro 474
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE 469	BS_SDK_ERROR_UNSUPPORTED_MEDIUM 474
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_PROJECTTYPE macro 469	BS_SDK_ERROR_UNSUPPORTED_MEDIUM macro 474
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION 470	BS_SDK_ERROR_VOLUME_OVERFLOW 474
BS_SDK_ERROR_NOT_ALLOWED_FOR_THIS_UDF_VERSION macro 470	BS_SDK_ERROR_VOLUME_OVERFLOW macro 474
BS_SDK_ERROR_NOT_IMPLEMENTED 470	BS_SDK_INT_ERROR_1 475
BS_SDK_ERROR_NOT_IMPLEMENTED macro 470	BS_SDK_INT_ERROR_1 macro 475
BS_SDK_ERROR_NOTREADY 470	BS_SDK_INT_ERROR_2 475
BS_SDK_ERROR_NOTREADY macro 470	BS_SDK_INT_ERROR_2 macro 475
BS_SDK_ERROR_NOTSUPPORTED 471	BS_SDK_INT_ERROR_3 475
BS_SDK_ERROR_NOTSUPPORTED macro 471	BS_SDK_INT_ERROR_3 macro 475
BS_SDK_ERROR_PARITYERR 471	BS_SDK_INT_ERROR_4 475
BS_SDK_ERROR_PARITYERR macro 471	BS_SDK_INT_ERROR_4 macro 475
BS_SDK_ERROR_PATH_EXISTS 471	BS_SDK_INT_ERROR_5 476
BS_SDK_ERROR_PATH_EXISTS macro 471	BS_SDK_INT_ERROR_5 macro 476
BS_SDK_ERROR_PLUGIN 471	BS_SDK_INT_ERROR_FORMAT 476
BS_SDK_ERROR_PLUGIN macro 471	BS_SDK_INT_ERROR_FORMAT macro 476
BS_SDK_ERROR_RESERVED 472	BS_SDK_KEY_INVALID 476
BS_SDK_ERROR_RESERVED macro 472	BS_SDK_KEY_INVALID macro 476
BS_SDK_ERROR_SELECTIONTIMEOUT 472	BS_SDK_KEY_VALID 476
BS_SDK_ERROR_SELECTIONTIMEOUT macro 472	BS_SDK_KEY_VALID macro 476
BS_SDK_ERROR_SRBTIMEOUT 472	BS_SDK_MESSAGE_01 477
BS_SDK_ERROR_SRBTIMEOUT macro 472	BS_SDK_MESSAGE_01 macro 477
BS_SDK_ERROR_TIMEOUT 472	BS_SDK_MESSAGE_02 477
BS_SDK_ERROR_TIMEOUT macro 472	BS_SDK_MESSAGE_02 macro 477
BS_SDK_ERROR_TOOMUCHDATA 473	BS_SDK_MESSAGE_03 477
BS_SDK_ERROR_TOOMUCHDATA macro 473	BS_SDK_MESSAGE_03 macro 477
BS_SDK_ERROR_TOOMUCHINDEXES 473	BS_SDK_MESSAGE_04 478
BS_SDK_ERROR_TOOMUCHINDEXES macro 473	BS_SDK_MESSAGE_04 macro 478
BS_SDK_ERROR_UNEXPECTEDBUSFREE 473	BS_SDK_MESSAGE_05 478
BS_SDK_ERROR_UNEXPECTEDBUSFREE macro 473	BS_SDK_MESSAGE_05 macro 478
BS_SDK_ERROR_UNITATTENTION 473	BS_SDK_MESSAGE_06 478
BS_SDK_ERROR_UNITATTENTION macro 473	BS_SDK_MESSAGE_06 macro 478
BS_SDK_ERROR_UNKNOWN 474	BS_SDK_MESSAGE_07 479
BS_SDK_ERROR_UNKNOWN macro 474	BS_SDK_MESSAGE_07 macro 479
	BS_SDK_MESSAGE_08 479
	BS_SDK_MESSAGE_08 macro 479
	BS_SDK_MESSAGE_10 479
	BS_SDK_MESSAGE_10 macro 479
	BS_SDK_MESSAGE_11 480
	BS_SDK_MESSAGE_11 macro 480

BS_SDK_MESSAGE_12 480	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES 486
BS_SDK_MESSAGE_12 macro 480	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_MATCHING_ENTRIES macro 486
BS_SDK_MESSAGE_13 480	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY 486
BS_SDK_MESSAGE_13 macro 480	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY macro 486
BS_SDK_MESSAGE_14 481	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY 486
BS_SDK_MESSAGE_14 macro 481	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY macro 486
BS_SDK_MESSAGE_15 481	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY 486
BS_SDK_MESSAGE_15 macro 481	BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_GETTING_SELECTED_ENTRY macro 486
BS_SDK_MESSAGE_16 481	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY 487
BS_SDK_MESSAGE_COMPENC_BOTH 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY macro 487
BS_SDK_MESSAGE_COMPENC_BOTH macro 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY macro 487
BS_SDK_MESSAGE_COMPENC_COMPRESSED 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY macro 487
BS_SDK_MESSAGE_COMPENC_COMPRESSED macro 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_FETCHING_SELECTED_ENTRY macro 487
BS_SDK_MESSAGE_COMPENC_ENCRYPTED 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES 487
BS_SDK_MESSAGE_COMPENC_ENCRYPTED macro 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES macro 487
BS_SDK_MESSAGE_ERASESTART 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES macro 487
BS_SDK_MESSAGE_ERASESTART macro 482	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_ENTRIES macro 487
BS_SDK_MESSAGE_EXTR_FILE 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS 488
BS_SDK_MESSAGE_EXTR_FILE macro 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_GETTING_MATCHING_PARAMS macro 488
BS_SDK_MESSAGE_FORMAT 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE 488
BS_SDK_MESSAGE_FORMAT macro 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE macro 488
BS_SDK_MESSAGE_FORMAT_DONE 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE 488
BS_SDK_MESSAGE_FORMAT_DONE macro 483	BS_SDK_MESSAGE_INTERNETDB_STATUS_INIT_COMPLETE macro 488
BS_SDK_MESSAGE_IMAGECREATESTART 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES 489
BS_SDK_MESSAGE_IMAGECREATESTART macro 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_MULTIPLE_MATCHES macro 489
BS_SDK_MESSAGE_IMPORT 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY 489
BS_SDK_MESSAGE_IMPORT macro 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY macro 489
BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY macro 489
BS_SDK_MESSAGE_INTERNETDB_STATUS_CANNOT_SELECT_ENTRY macro 484	BS_SDK_MESSAGE_INTERNETDB_STATUS_SELECT_ENTRY macro 489
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CR EATING_MATCHING_PARAM 485	BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH 490
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CR EATING_MATCHING_PARAM macro 485	BS_SDK_MESSAGE_INTERNETDB_STATUS_SINGLE_MATCH macro 490
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CR EATING_PROVIDER 485	BS_SDK_MESSAGE_SIMULATE 490
BS_SDK_MESSAGE_INTERNETDB_STATUS_ERROR_CR EATING_PROVIDER macro 485	

BS_SDK_MESSAGE_SIMULATE macro 490	BS_TCI_FRAME_OFFSET 495
BS_SDK_MESSAGE_WAIT 490	BS_TCI_FRAME_OFFSET macro 495
BS_SDK_MESSAGE_WAIT macro 490	BS_TCI_GENRE 496
BS_SDK_MESSAGE_WRITESTART 491	BS_TCI_GENRE macro 496
BS_SDK_MESSAGE_WRITESTART macro 491	BS_TCI_REVISION 496
BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE 491	BS_TCI_REVISION macro 496
BS_SDK_VERIFY_ERROR_CDFILEUNREADABLE macro 491	BS_TCI_SUBMITTED_VIA 496
BS_SDK_VERIFY_ERROR_FILESDIFFERENT 491	BS_TCI_SUBMITTED_VIA macro 496
BS_SDK_VERIFY_ERROR_FILESDIFFERENT macro 491	BS_TCI_TITLE 496
BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE 491	BS_TCI_TITLE macro 496
BS_SDK_VERIFY_ERROR_HDDFILEUNREADABLE macro 491	BS_TCI_YEAR 497
BS_SEPARATE_CHANNEL_MUTE 492	BS_TCI_YEAR macro 497
BS_SEPARATE_CHANNEL_MUTE macro 492	BS_TF_AUDIO 497
BS_SEPARATE_VOLUME_LEVELS 492	BS_TF_AUDIO macro 497
BS_SEPARATE_VOLUME_LEVELS macro 492	BS_TF_DATA_MODE1 497
BS_SMART 492	BS_TF_DATA_MODE1 macro 497
BS_SMART macro 492	BS_TF_DATA_MODE2 497
BS_STREAMING 492	BS_TF_DATA_MODE2 macro 497
BS_STREAMING macro 492	BS_TM_DMA 498
BS_STRING_MANIPULATION_DETECTED 493	BS_TM_DMA macro 498
BS_STRING_MANIPULATION_DETECTED macro 493	BS_TM_NOT_APPLICABLE 498
BS_TCH_CDTEXT 493	BS_TM_NOT_APPLICABLE macro 498
BS_TCH_CDTEXT macro 493	BS_TM_PIO 498
BS_TCH_CDTEXT_INTERNETDB 493	BS_TM_PIO macro 498
BS_TCH_CDTEXT_INTERNETDB macro 493	BS_TM_UNKNOWN 498
BS_TCH_INTERNETDB 493	BS_TM_UNKNOWN macro 498
BS_TCH_INTERNETDB macro 493	BS_TRUE 499
BS_TCH_INTERNETDB_CDTEXT 494	BS_TRUE macro 499
BS_TCH_INTERNETDB_CDTEXT macro 494	BS_UDF_PARTITION_PHYSICAL 499
BS_TCH_NONE 494	BS_UDF_PARTITION_PHYSICAL macro 499
BS_TCH_NONE macro 494	BS_UDF_PARTITION_SPARABLE 499
BS_TCI_ARTIST 494	BS_UDF_PARTITION_SPARABLE macro 499
BS_TCI_ARTIST macro 494	BS_UDF_PARTITION_VIRTUAL 500
BS_TCI_CATEGORY 494	BS_UDF_PARTITION_VIRTUAL macro 500
BS_TCI_CATEGORY macro 494	BS_UDF_VERSION_102 500
BS_TCI_DISK_LENGTH 495	BS_UDF_VERSION_102 macro 500
BS_TCI_DISK_LENGTH macro 495	BS_UDF_VERSION_150 500
BS_TCI_DISKID 495	BS_UDF_VERSION_150 macro 500
BS_TCI_DISKID macro 495	BS_UDF_VERSION_200 501
BS_TCI_EXTENDED_INFO 495	BS_UDF_VERSION_200 macro 501
BS_TCI_EXTENDED_INFO macro 495	BS_UDF_VERSION_201 501
	BS_UDF_VERSION_201 macro 501

BS_UDF_VERSION_250	501	BS_WM.DAO96	506	
BS_UDF_VERSION_250	macro	501	BS_WM.DAO96 macro	506
BS_UDF_VERSION_260	501	BS_WM.TAO	507	
BS_UDF_VERSION_260	macro	501	BS_WM.TAO macro	507
BS_UDFOPT_ALL	502	BS_WRITE_BLURAY_R	507	
BS_UDFOPT_ALL	macro	502	BS_WRITE_BLURAY_R macro	507
BS_UDFOPT_AVCHD_DISC	502	BS_WRITE_BLURAY_R_XL	507	
BS_UDFOPT_AVCHD_DISC	macro	502	BS_WRITE_BLURAY_R_XL macro	507
BS_UDFOPT_FILE_STREAMS	502	BS_WRITE_BLURAY_RE	507	
BS_UDFOPT_FILE_STREAMS	macro	502	BS_WRITE_BLURAY_RE macro	507
BS_UDFOPT_IMPLEMENTATION_ID	502	BS_WRITE_BLURAY_RE_XL	508	
BS_UDFOPT_IMPLEMENTATION_ID	macro	502	BS_WRITE_BLURAY_RE_XL macro	508
BS_UDFOPT_PARTITION_TYPE	503	BS_WRITE_CDR	508	
BS_UDFOPT_PARTITION_TYPE	macro	503	BS_WRITE_CDR macro	508
BS_UDFOPT_VERSION	503	BS_WRITE_CDRW	508	
BS_UDFOPT_VERSION	macro	503	BS_WRITE_CDRW macro	508
BS_UNDERRUN_PROTECTION	503	BS_WRITE_CDRW_CAV	508	
BS_UNDERRUN_PROTECTION	macro	503	BS_WRITE_CDRW_CAV macro	508
BS_UNKNOWN	503	BS_WRITE_DVD_DL	509	
BS_UNKNOWN	macro	503	BS_WRITE_DVD_DL macro	509
BS_UPC_READ	504	BS_WRITE_DVD_MRDL	509	
BS_UPC_READ	macro	504	BS_WRITE_DVD_MRDL macro	509
BS_VCD_INFINITE	504	BS_WRITE_DVD_RDL_PLUS	509	
BS_VCD_INFINITE	macro	504	BS_WRITE_DVD_RDL_PLUS macro	509
BS_VCD_KEY_0	504	BS_WRITE_DVD_RWDL_PLUS	509	
BS_VCD_KEY_0	macro	504	BS_WRITE_DVD_RWDL_PLUS macro	509
BS_VCD_KEY_DEFAULT	504	BS_WRITE_DVDR	510	
BS_VCD_KEY_DEFAULT	macro	504	BS_WRITE_DVDR macro	510
BS_VCD_KEY_NEXT	505	BS_WRITE_DVDR_PLUS	510	
BS_VCD_KEY_NEXT	macro	505	BS_WRITE_DVDR_PLUS macro	510
BS_VCD_KEY_PREVIOUS	505	BS_WRITE_DVDRAM	510	
BS_VCD_KEY_PREVIOUS	macro	505	BS_WRITE_DVDRAM macro	510
BS_VCD_KEY_RETURN	505	BS_WRITE_DVDRW	510	
BS_VCD_KEY_RETURN	macro	505	BS_WRITE_DVDRW macro	510
BS_VCD_SEGMENT_ITEM_0	505	BS_WRITE_DVDRW_PLUS	511	
BS_VCD_SEGMENT_ITEM_0	macro	505	BS_WRITE_DVDRW_PLUS macro	511
BS_VCD_TRACK_ITEM_0	506	BS_WRITE_HDDVD_R	511	
BS_VCD_TRACK_ITEM_0	macro	506	BS_WRITE_HDDVD_R macro	511
BS_WHOLE_PATH	506	BS_WRITE_HDDVD_RW	511	
BS_WHOLE_PATH	macro	506	BS_WRITE_HDDVD_RW macro	511
BS_WM.DAO	506	BS_WRITE_MOUNT_RAINER	511	
BS_WM.DAO	macro	506	BS_WRITE_MOUNT_RAINER macro	511

BS_WRITE_TEST 512	CreateDir function 127
BS_WRITE_TEST macro 512	CreateDirEvent 238
Burn 121	CreateDirEvent type 238
Burn function 121	CreateImage 127
BurnDialog 121	CreateImage function 127
BurnDialog function 121	CreateProject 128
BurnDoneEvent 237	CreateProject function 128
BurnDoneEvent type 237	
BurnFileEvent 237	<b>D</b>
BurnFileEvent type 237	DABurn 129
BurnISO 122	DABurn function 129
BurnISO function 122	DABurnISOImage 129
	DABurnISOImage function 129
<b>C</b>	DACheckSignature 130
CheckLicenseKey 123	DACheckSignature function 130
CheckLicenseKey function 123	DACloseDevice 130
CheckSignature 123	DACloseDevice function 130
CheckSignature function 123	DACloseSession 130
Classes 1	DACloseSession function 130
ClearAll 123	DAConvertSpeedFromKBPerSec 131
ClearAll function 123	DAConvertSpeedFromKBPerSec function 131
CloseCDTextHandle 124	DACopyDisk 131
CloseCDTextHandle function 124	DACopyDisk function 131
CloseDevice 124	DACreateImage 132
CloseDevice function 124	DACreateImage function 132
CloseDirectory 124	DAEjectDevice 132
CloseDirectory function 124	DAEjectDevice function 132
CloseDiskSession 125	DAErase 133
CloseDiskSession function 125	DAErase function 133
CloseNetworkTagsHandle 125	DAGetBurnSpeed 133
CloseNetworkTagsHandle function 125	DAGetBurnSpeed function 133
CloseSession 126	DAGetDeviceCapabilities 134
CloseSession function 126	DAGetDeviceCapabilities function 134
CompareFilesForArrangementEvent 237	DAGetDeviceCapabilitiesHandle 134
CompareFilesForArrangementEvent type 237	DAGetDeviceCapabilitiesHandle function 134
COMPRESS_BLOCKSIZE 512	DAGetDeviceInformation 135
COMPRESS_BLOCKSIZE macro 512	DAGetDeviceInformation function 135
ConvertSpeedFromKBPerSec 126	DAGetDeviceInformationEx 135
ConvertSpeedFromKBPerSec function 126	DAGetDeviceInformationEx function 135
CopyDisk 127	DAGetMaxBurnSpeed 136
CopyDisk function 127	DAGetMaxBurnSpeed function 136
CreateDir 127	DAGetMaxReadSpeed 136

DAGetMaxReadSpeed function 136  
DAGetMediumFreedBld 137  
DAGetMediumFreedBld function 137  
DAGetMediumInformation 137  
DAGetMediumInformation function 137  
DAGetPossibleBurnSpeeds 138  
DAGetPossibleBurnSpeeds function 138  
DAGetPossibleImageFormats 138  
DAGetPossibleImageFormats function 138  
DAGetPossibleReadSpeeds 139  
DAGetPossibleReadSpeeds function 139  
DAGetReadSpeed 139  
DAGetReadSpeed function 139  
DAGetSessionInformation 140  
DAGetSessionInformation function 140  
DAGetTrackFormatEx 140  
DAGetTrackFormatEx function 140  
DAGetTrackIndexes 141  
DAGetTrackIndexes function 141  
DAGetTrackInformation 141  
DAGetTrackInformation function 141  
DAGetTrackISRC 142  
DAGetTrackISRC function 142  
DAGrabAudioTrack 142  
DAGrabAudioTrack function 142  
DAImportFile 143  
DAImportFile function 143  
DAImportFileEx 143  
DAImportFileEx function 143  
DAIsDeviceReady 144  
DAIsDeviceReady function 144  
DALockMedium 144  
DALockMedium function 144  
DAOpenDiskSession 145  
DAOpenDiskSession function 145  
DAPlayAudioTrack 145  
DAPlayAudioTrack function 145  
DAPrepare 146  
DAPrepare function 146  
DAReadCDText 146  
DAReadCDText function 146  
DAReadFileContents 147  
DAReadFileContents function 147  
DAReadSectors 147  
DAReadSectors function 147  
DASaveTrackToFile 148  
DASaveTrackToFile function 148  
DASetBurnSpeed 148  
DASetBurnSpeed function 148  
DASetReadSpeed 149  
DASetReadSpeed function 149  
DASetRegionalCode 149  
DASetRegionalCode function 149  
DAVerifyFile 150  
DAVerifyFile function 150  
DelInitALL 150  
DelInitALL function 150  
DelInitialize 151  
DelInitialize function 151  
DeleteProject 151  
DeleteProject function 151  
DirExists 151  
DirExists function 151

## E

EjectDevice 152  
EjectDevice function 152  
EnableImageDevice 152  
EnableImageDevice function 152  
EnableMCNDisabling 153  
EnableMCNDisabling function 153  
ENCRYPT\_BLOCKSIZE 512  
ENCRYPT\_BLOCKSIZE macro 512  
Erase 153  
Erase function 153  
EraseDialog 154  
EraseDialog function 154  
EraseDoneEvent 238  
EraseDoneEvent type 238  
EraseMpegByIndex 154  
EraseMpegByIndex function 154

## F

Files 516

FinalizeEvent 239	GetCompressEncrypt function 161
FinalizeEvent type 239	GetDeviceCapabilities 162
FOpen 512	GetDeviceCapabilities function 162
FOpen macro 512	GetDeviceCapabilitiesHandle 162
ForceDeInitialize 154	GetDeviceCapabilitiesHandle function 162
ForceDeInitialize function 154	GetDeviceInformation 163
FoxEntries 233	GetDeviceInformation function 163
FoxEntries structure 233	GetDeviceInformationEx 163
FoxIndecies 233	GetDeviceInformationEx function 163
FoxIndecies structure 233	GetDevices 164
Functions 108	GetDevices function 164
	GetDVDVideoOptions 164
	GetDVDVideoOptions function 164

## G

GetActiveDevicesCount 155	GetEraseDoneEventCallback 165
GetActiveDevicesCount function 155	GetEraseDoneEventCallback function 165
GetASPI 155	GetErrorDeviceName 165
GetASPI function 155	GetErrorDeviceName function 165
GetAudioFileSize 156	GetFileAllocationTable 165
GetAudioFileSize function 156	GetFileAllocationTable function 165
GetAudioTags 156	GetFileEntry 166
GetAudioTags function 156	GetFileEntry function 166
GetBootInfoEx 156	GetFileTimeEx 166
GetBootInfoEx function 156	GetFileTimeEx function 166
GetBootVolumeInformation 157	GetFinalizeEventCallback 167
GetBootVolumeInformation function 157	GetFinalizeEventCallback function 167
GetBurnDevice 157	GetImagePath 167
GetBurnDevice function 157	GetImagePath function 167
GetBurnDevices 158	GetImageSize 168
GetBurnDevices function 158	GetImageSize function 168
GetBurnDoneEventCallback 158	GetInfoTextEventCallback 168
GetBurnDoneEventCallback function 158	GetInfoTextEventCallback function 168
GetBurnFileEventCallback 159	GetISOInfoEx 168
GetBurnFileEventCallback function 159	GetISOInfoEx function 168
GetBurnSpeed 159	GetISOVolumeInformation 169
GetBurnSpeed function 159	GetISOVolumeInformation function 169
GetCDTextDiskTagString 160	GetJobDoneEventCallback 169
GetCDTextDiskTagString function 160	GetJobDoneEventCallback function 169
GetCDTextTrackTagString 160	GetLanguage 170
GetCDTextTrackTagString function 160	GetLanguage function 170
GetCompareFilesForArrangementEventCallback 161	GetMaxBurnSpeed 170
GetCompareFilesForArrangementEventCallback function 161	GetMaxBurnSpeed function 170
GetCompressEncrypt 161	GetMaxReadSpeed 171

GetMaxReadSpeed function 171	GetStartVerifyEventCallback function 181
GetMediumFreedbld 171	GetText 181
GetMediumFreedbld function 171	GetText function 181
GetMediumInformation 172	GetTextEvent 239
GetMediumInformation function 172	GetTextEvent type 239
GetMpegCount 172	GetTmpPath 182
GetMpegCount function 172	GetTmpPath function 182
GetNetworkDiskTagInt 172	GetTrackFormatEx 182
GetNetworkDiskTagInt function 172	GetTrackFormatEx function 182
GetNetworkDiskTagString 173	GetTrackIndexes 183
GetNetworkDiskTagString function 173	GetTrackIndexes function 183
GetNetworkTrackTagInt 174	GetTrackInformation 183
GetNetworkTrackTagInt function 174	GetTrackInformation function 183
GetNetworkTrackTagString 174	GetTrackISRC 184
GetNetworkTrackTagString function 174	GetTrackISRC function 184
GetNumberOfFiles 175	GetUDFOptions 184
GetNumberOfFiles function 175	GetUDFOptions function 184
GetOptions 175	GetUDFOptionsEx 185
GetOptions function 175	GetUDFOptionsEx function 185
GetPlayTime 176	GetUDFVolumeInformation 185
GetPlayTime function 176	GetUDFVolumeInformation function 185
GetPossibleBurnSpeeds 176	GetVerify 185
GetPossibleBurnSpeeds function 176	GetVerify function 185
GetPossibleImageFormats 177	GetVerifyDoneEventCallback 186
GetPossibleImageFormats function 177	GetVerifyDoneEventCallback function 186
GetPossibleReadSpeeds 177	GetVerifyErrorEventCallback 186
GetPossibleReadSpeeds function 177	GetVerifyErrorEventCallback function 186
GetPrecisePlayTime 178	GetVerifyFileEventCallback 187
GetPrecisePlayTime function 178	GetVerifyFileEventCallback function 187
GetProcessEventCallback 178	GetVerifySectorEventCallback 187
GetProcessEventCallback function 178	GetVerifySectorEventCallback function 187
GetProjectType 178	GetWriteCDTextInUnicode 187
GetProjectType function 178	GetWriteCDTextInUnicode function 187
GetRAWDataEventCallback 179	GrabAudioTrack 188
GetRAWDataEventCallback function 179	GrabAudioTrack function 188
GetReadDevice 179	<b>H</b>
GetReadDevice function 179	HDIR 239
GetReadSpeed 180	HDIR type 239
GetReadSpeed function 180	HSESSION 240
GetSessionInformation 180	HSESSION type 240
GetSessionInformation function 180	
GetStartVerifyEventCallback 181	

LockMedium function 191

## M

ImportFile 188  
ImportFile function 188  
ImportFileEx 189  
ImportFileEx function 189  
InfoTextEvent 240  
InfoTextEvent type 240  
Initialize 189  
Initialize function 189  
INSERT\_STRUCTURE\_PADDING 513  
INSERT\_STRUCTURE\_PADDING macro 513  
int16 240  
int16 type 240  
int2bool 513  
int2bool macro 513  
int2BS\_BOOL 513  
int2BS\_BOOL macro 513  
int32 241  
int32 type 241  
int64 241  
int64 type 241  
int8 241  
int8 type 241  
IsDeviceReady 190  
IsDeviceReady function 190  
IsoSDKBurningLib.h 517  
IsoSDKDefinitions.h 517  
IsoSDKEExport.h 577  
IsoSDKUnicode.h 587  
IsValidVideoTsFolder 190  
IsValidVideoTsFolder function 190

## J

JobDoneEvent 241  
JobDoneEvent type 241

## L

LoadBassPlugin 191  
LoadBassPlugin function 191  
LockMedium 191

## O

OpenDirectory 192  
OpenDirectory function 192  
OpenDiskSession 193  
OpenDiskSession function 193

## P

PATHSEPSTRING 513  
PATHSEPSTRING macro 513  
PFoxEntries 234  
PFoxEntries structure 234  
PFoxIndecies 234  
PFoxIndecies structure 234  
PlayAudioFile 193  
PlayAudioFile function 193  
PlayAudioTrack 194  
PlayAudioTrack function 194  
Prepare 194  
Prepare function 194  
ProcessEvent 242  
ProcessEvent type 242

## R

RAWDataEvent 242  
RAWDataEvent type 242  
ReadCDText 194  
ReadCDText function 194  
ReadDirectory 195  
ReadDirectory function 195  
ReadFileContents 195  
ReadFileContents function 195  
ReadSectors 196  
ReadSectors function 196  
ReleaseDeviceCapabilities 196

ReleaseDeviceCapabilities function 196  
RemoveBurnDevice 197  
RemoveBurnDevice function 197  
RemoveDir 197  
RemoveDir function 197  
RemoveFile 198  
RemoveFile function 198  
RemoveFileEvent 243  
RemoveFileEvent type 243  
RenameDir 198  
RenameDir function 198  
RenameFile 198  
RenameFile function 198  
RescanDevices 199  
RescanDevices function 199  
ResetCallbacks 199  
ResetCallbacks function 199

**S**

SAudioGrabbingParams 2  
SAudioGrabbingParams structure 2  
    about SAUDIOGRABBINGPARAMS structure 2  
    INSERT\_STRUCTURE\_PADDING 6  
    nBitrate 4  
    nBitrateType 4  
    nEncoderType 5  
    nMaxBitrate 5  
    nMinBitrate 5  
    nNetworkTagsHandle 5  
    nQuality 5  
    nTagChoice 6  
    SAudioGrabbingParams 3  
    SAudioGrabbingParams data members 3  
    SAudioGrabbingParams members 2  
    SAudioGrabbingParams methods 6  
SAudioGrabbingParams::INSERT\_STRUCTURE\_PADDING 6  
SAudioGrabbingParams::nBitrate 4  
SAudioGrabbingParams::nBitrateType 4  
SAudioGrabbingParams::nEncoderType 5  
SAudioGrabbingParams::nMaxBitrate 5  
SAudioGrabbingParams::nMinBitrate 5  
SAudioGrabbingParams::nNetworkTagsHandle 5  
SAudioGrabbingParams::nQuality 5  
SAudioGrabbingParams::nTagChoice 6  
SAudioGrabbingParams::SAudioGrabbingParams 3  
SaveLogToFile 200  
SaveLogToFile function 200  
SaveOptionsToFile 200  
SaveOptionsToFile function 200  
SaveTrackToFile 201  
SaveTrackToFile function 201  
SBootInfoEx 6  
SBootInfoEx structure 6  
    about SBootInfoEx structure 6  
    BootIndicator 8  
    DeveloperID 8  
    Emulation 8  
    INSERT\_STRUCTURE\_PADDING 9  
    LoadSegment 9  
    PlatformID 9  
    SBootInfoEx 7  
    SBootInfoEx data members 7  
    SBootInfoEx members 7  
    SBootInfoEx methods 9  
    SectorCount 9  
    SBootInfoEx::BootIndicator 8  
    SBootInfoEx::DeveloperID 8  
    SBootInfoEx::Emulation 8  
    SBootInfoEx::INSERT\_STRUCTURE\_PADDING 9  
    SBootInfoEx::LoadSegment 9  
    SBootInfoEx::PlatformID 9  
    SBootInfoEx::SBootInfoEx 7  
    SBootInfoEx::SectorCount 9  
    SBootVolumeInfo 10  
    SBootVolumeInfo structure 10  
        about SBootVolumeInfo structure 10  
        INSERT\_STRUCTURE\_PADDING 11  
        nVolumeDescriptorAddress 11  
        SBootVolumeInfo 10  
        SBootVolumeInfo data members 10  
        SBootVolumeInfo members 10  
        SBootVolumeInfo methods 11  
        sInfoEx 11  
    SBootVolumeInfo::INSERT\_STRUCTURE\_PADDING 11

SBootVolumeInfo::nVolumeDescriptorAddress 11  
SBootVolumeInfo::SBootVolumeInfo 10  
SBootVolumeInfo::sInfoEx 11  
SCompressEncryptInfo 11  
SCompressEncryptInfo structure 11  
    about SCompressEncryptInfo structure 11  
    bCompression 13  
    bEncryption 13  
    INSERT\_STRUCTURE\_PADDING 13  
    lpszPassword 13  
    nCompressionLevel 13  
    SCompressEncryptInfo 12  
    SCompressEncryptInfo data members 12  
    SCompressEncryptInfo members 12  
    SCompressEncryptInfo methods 13  
SCompressEncryptInfo::bCompression 13  
SCompressEncryptInfo::bEncryption 13  
SCompressEncryptInfo::INSERT\_STRUCTURE\_PADDING 13  
    SCompressEncryptInfo::lpszPassword 13  
    SCompressEncryptInfo::nCompressionLevel 13  
    SCompressEncryptInfo::SCompressEncryptInfo 12  
SCreateImageParams 14  
SCreateImageParams structure 14  
    about SCreateImageParams structure 14  
    bFullCapacity 15  
    cErrorParams 15  
    INSERT\_STRUCTURE\_PADDING 16  
    lpszBadSectorsFilePath 15  
    lpszImagePath 15  
    nImageType 16  
    nVerifyBufferSectors 16  
    SCreateImageParams 14  
    SCreateImageParams data members 15  
    SCreateImageParams members 14  
    SCreateImageParams methods 16  
SCreateImageParams::bFullCapacity 15  
SCreateImageParams::cErrorParams 15  
SCreateImageParams::INSERT\_STRUCTURE\_PADDING 16  
SCreateImageParams::lpszBadSectorsFilePath 15  
SCreateImageParams::lpszImagePath 15  
SCreateImageParams::nImageType 16  
SCreateImageParams::nVerifyBufferSectors 16  
SCreateImageParams::SCreateImageParams 14  
SDirToAdd 16  
SDirToAdd structure 16  
    about SDirToAdd structure 16  
    bRecursive 18  
    INSERT\_STRUCTURE\_PADDING 19  
    lpszDestinationPath 18  
    lpszFileSpecification 18  
    lpszSourceDirPath 18  
    nFileAttributes 19  
    nSavePath 19  
    SDirToAdd 17  
    SDirToAdd data members 17  
    SDirToAdd members 17  
    SDirToAdd methods 19  
SDirToAdd::bRecursive 18  
SDirToAdd::INSERT\_STRUCTURE\_PADDING 19  
SDirToAdd::lpszDestinationPath 18  
SDirToAdd::lpszFileSpecification 18  
SDirToAdd::lpszSourceDirPath 18  
SDirToAdd::nFileAttributes 19  
SDirToAdd::nSavePath 19  
SDirToAdd::SDirToAdd 17  
SDirToCreate 20  
SDirToCreate structure 20  
    about SDdirToCreate structure 20  
    INSERT\_STRUCTURE\_PADDING 21  
    lpszDestinationPath 21  
    lpszDir 21  
    SDirToCreate 20  
    SDirToCreate data members 20  
    SDirToCreate members 20  
    SDirToCreate methods 21  
SDirToCreate::INSERT\_STRUCTURE\_PADDING 21  
SDirToCreate::lpszDestinationPath 21  
SDirToCreate::lpszDir 21  
SDirToCreate::SDirToCreate 20  
SDirToRemove 21  
SDirToRemove structure 21  
    about SDirToRemove structure 21  
    INSERT\_STRUCTURE\_PADDING 22  
    lpszDestinationPath 22

SDirToRemove 22  
SDirToRemove data members 22  
SDirToRemove members 21  
SDirToRemove methods 22  
SDirToRemove::INSERT\_STRUCTURE\_PADDING 22  
SDirToRemove::lpszDestinationPath 22  
SDirToRemove::SDirToRemove 22  
SDirToRename 23  
SDirToRename structure 23  
    about SDirToRename structure 23  
    INSERT\_STRUCTURE\_PADDING 24  
    lpszDestinationPath 24  
    lpszSourcePath 24  
    SDirToRename 23  
    SDirToRename data members 23  
    SDirToRename members 23  
    SDirToRename methods 24  
SDirToRename::INSERT\_STRUCTURE\_PADDING 24  
SDirToRename::lpszDestinationPath 24  
SDirToRename::lpszSourcePath 24  
SDirToRename::SDirToRename 23  
SDiskCopyOptions 24  
SDiskCopyOptions structure 24  
    about SDiskCopyOptions structure 24  
    bEjectAfterBurn 26  
    bFullCapacity 26  
    bVerifyAfterBurn 26  
    cErrorParams 27  
    INSERT\_STRUCTURE\_PADDING 27  
    nReadMode 27  
    nVerifyBufferSectors 27  
    nWriteMethod 27  
    SDiskCopyOptions 25  
    SDiskCopyOptions data members 25  
    SDiskCopyOptions members 25  
    SDiskCopyOptions methods 27  
SDiskCopyOptions::bEjectAfterBurn 26  
SDiskCopyOptions::bFullCapacity 26  
SDiskCopyOptions::bVerifyAfterBurn 26  
SDiskCopyOptions::cErrorParams 27  
SDiskCopyOptions::INSERT\_STRUCTURE\_PADDING 27  
SDiskCopyOptions::nReadMode 27  
SDiskCopyOptions::nVerifyBufferSectors 27  
SDiskCopyOptions::nWriteMethod 27  
SDiskCopyOptions::SDiskCopyOptions 25  
SDVDVideoOptions 28  
SDVDVideoOptions structure 28  
    about SDVDVideoOptions structure 28  
    ForceUppercase 29  
    INSERT\_STRUCTURE\_PADDING 29  
    Padding 29  
    SDVDVideoOptions 28  
    SDVDVideoOptions data members 28  
    SDVDVideoOptions members 28  
    SDVDVideoOptions methods 29  
SDVDVideoOptions::ForceUppercase 29  
SDVDVideoOptions::INSERT\_STRUCTURE\_PADDING 29  
SDVDVideoOptions::Padding 29  
SDVDVideoOptions::SDVDVideoOptions 28  
SetAddFileEventCallback 201  
SetAddFileEventCallback function 201  
SetASPI 202  
SetASPI function 202  
SetAudioDecodeDoneEventCallback 203  
SetAudioDecodeDoneEventCallback function 203  
SetAudioDecoderEventCallback 203  
SetAudioDecoderEventCallback function 203  
SetAudioFileProperty 204  
SetAudioFileProperty function 204  
SetBootInfoEx 204  
SetBootInfoEx function 204  
SetBurnDevice 205  
SetBurnDevice function 205  
SetBurnDoneEventCallback 205  
SetBurnDoneEventCallback function 205  
SetBurnFileEventCallback 206  
SetBurnFileEventCallback function 206  
SetBurnSpeed 206  
SetBurnSpeed function 206  
SetCompareFilesForArrangementEventCallback 207  
SetCompareFilesForArrangementEventCallback function 207  
SetCompressEncrypt 208  
SetCompressEncrypt function 208  
SetCreateDirEventCallback 208

SetCreateDirEventCallback function 208	SetReadDevice function 219
SetDVDVideoOptions 209	SetReadSpeed 219
SetDVDVideoOptions function 209	SetReadSpeed function 219
SetEraseDoneEventCallback 209	SetRegionalCode 220
SetEraseDoneEventCallback function 209	SetRegionalCode function 220
SetFileAttr 210	SetRemoveFileEventCallback 221
SetFileAttr function 210	SetRemoveFileEventCallback function 221
SetFileTimeEx 210	SetStartVerifyEventCallback 221
SetFileTimeEx function 210	SetStartVerifyEventCallback function 221
SetFileTimes 211	SetTmpPath 222
SetFileTimes function 211	SetTmpPath function 222
SetFileUserParam 211	SetUDFOptions 222
SetFileUserParam function 211	SetUDFOptions function 222
SetFinalizeEventCallback 212	SetUDFOptionsEx 223
SetFinalizeEventCallback function 212	SetUDFOptionsEx function 223
SetFXApp 212	SetVCDKeyHandler 223
SetFXApp function 212	SetVCDKeyHandler function 223
SetGetTextEventCallback 213	SetVCDTimeOutHandler 224
SetGetTextEventCallback function 213	SetVCDTimeOutHandler function 224
SetIgnoreFileExist 213	SetVerify 224
SetIgnoreFileExist function 213	SetVerify function 224
SetImagePath 214	SetVerifyDoneEventCallback 224
SetImagePath function 214	SetVerifyDoneEventCallback function 224
SetInfoTextEventCallback 214	SetVerifyErrorEventCallback 225
SetInfoTextEventCallback function 214	SetVerifyErrorEventCallback function 225
SetISOInfoEx 215	SetVerifyFileEventCallback 226
SetISOInfoEx function 215	SetVerifyFileEventCallback function 226
SetJobDoneEventCallback 215	SetVerifySectorEventCallback 226
SetJobDoneEventCallback function 215	SetVerifySectorEventCallback function 226
SetLanguage 216	SetVideoScanDoneEventCallback 227
SetLanguage function 216	SetVideoScanDoneEventCallback function 227
SetOptions 217	SetVideoScannerEventCallback 228
SetOptions function 217	SetVideoScannerEventCallback function 228
SetOptionsFromFile 217	SetWriteCDTextInUnicode 228
SetOptionsFromFile function 217	SetWriteCDTextInUnicode function 228
SetProcessEventCallback 217	SExtendedDeviceInformation 29
SetProcessEventCallback function 217	SExtendedDeviceInformation structure 29
SetRAWDataEventCallback 218	about SExtendedDeviceInformation structure 29
SetRAWDataEventCallback function 218	FirmwareCreationDate 31
SetRAWStructure 219	INSERT_STRUCTURE_PADDING 34
SetRAWStructure function 219	IpszConnectionInterface 32
SetReadDevice 219	IpszLoaderType 32

IpszName 32  
IpszPhysicalInterface 32  
IpszRevision 32  
IpszSerialNumber 32  
nBufferSize 33  
nDeTransferMode 33  
nNumberOfVolumeLevels 33  
nReadRetryCount 33  
nRegionCode 33  
nRegionCodeChangesLeft 33  
nRegionCodeVendorResetsLeft 34  
SExtendedDeviceInformation 31  
SExtendedDeviceInformation data members 31  
SExtendedDeviceInformation members 30  
SExtendedDeviceInformation methods 34  
SExtendedDeviceInformation::FirmwareCreationDate 31  
SExtendedDeviceInformation::INSERT\_STRUCTURE\_PADDI NG 34  
SExtendedDeviceInformation::IpszConnectionInterface 32  
SExtendedDeviceInformation::IpszLoaderType 32  
SExtendedDeviceInformation::IpszName 32  
SExtendedDeviceInformation::IpszPhysicalInterface 32  
SExtendedDeviceInformation::IpszRevision 32  
SExtendedDeviceInformation::IpszSerialNumber 32  
SExtendedDeviceInformation::nBufferSize 33  
SExtendedDeviceInformation::nDeTransferMode 33  
SExtendedDeviceInformation::nNumberOfVolumeLevels 33  
SExtendedDeviceInformation::nReadRetryCount 33  
SExtendedDeviceInformation::nRegionCode 33  
SExtendedDeviceInformation::nRegionCodeChangesLeft 33  
SExtendedDeviceInformation::nRegionCodeVendorResetsLeft 34  
SExtendedDeviceInformation::SExtendedDeviceInformation 31  
SExtent 231  
SExtent structure 231  
SFileAllocationTable 232  
SFileAllocationTable structure 232  
SFileAudioProperty 34  
SFileAudioProperty structure 34  
    about SFileAudioProperty structure 34  
Arranger 36  
bPauseInFrames 36  
Composer 36  
INSERT\_STRUCTURE\_PADDING 38  
IpszMCN\_ISRC 36  
IpszSourceFilePath 36  
IpszTitle 37  
Message 37  
nIndexesLength 37  
nPause 37  
Performer 37  
pIndexes 37  
SFileAudioProperty 35  
SFileAudioProperty data members 35  
SFileAudioProperty members 35  
SFileAudioProperty methods 38  
SongWriter 38  
SFileAudioProperty::Arranger 36  
SFileAudioProperty::bPauseInFrames 36  
SFileAudioProperty::Composer 36  
SFileAudioProperty::INSERT\_STRUCTURE\_PADDING 38  
SFileAudioProperty::IpszMCN\_ISRC 36  
SFileAudioProperty::IpszSourceFilePath 36  
SFileAudioProperty::IpszTitle 37  
SFileAudioProperty::Message 37  
SFileAudioProperty::nIndexesLength 37  
SFileAudioProperty::nPause 37  
SFileAudioProperty::Performer 37  
SFileAudioProperty::pIndexes 37  
SFileAudioProperty::SFileAudioProperty 35  
SFileAudioProperty::SongWriter 38  
SFileDateTime 38  
SFileDateTime structure 38  
    about SFileDateTime structure 38  
    INSERT\_STRUCTURE\_PADDING 40  
    nDay 39  
    nHour 39  
    nMinute 39  
    nMonth 39  
    nSecond 40  
    nYear 40  
    SFileDateTime data members 39  
    SFileDateTime members 38

SFileDateTime methods 40  
SFileDateTime::INSERT\_STRUCTURE\_PADDING 40  
SFileDateTime::nDay 39  
SFileDateTime::nHour 39  
SFileDateTime::nMinute 39  
SFileDateTime::nMonth 39  
SFileDateTime::nSecond 40  
SFileDateTime::nYear 40  
SFileEntry 40  
SFileEntry structure 40  
    about SFileEntry structure 40  
    cAccessTime 42  
    cCreationTime 42  
    cDateTime 42  
    INSERT\_STRUCTURE\_PADDING 44  
    INSERT\_STRUCTURE\_PADDING method 44  
    lpszFileName 42  
    lpszFileOrigin 42  
    lpszFilePath 43  
    nAddress 43  
    nAttrib 43  
    nFileSize 43  
    pUserParam 43  
    SFileEntry data members 41  
    SFileEntry members 41  
    SFileEntry methods 44  
SFileEntry::cAccessTime 42  
SFileEntry::cCreationTime 42  
SFileEntry::cDateTime 42  
SFileEntry::INSERT\_STRUCTURE\_PADDING 44  
SFileEntry::lpszFileName 42  
SFileEntry::lpszFileOrigin 42  
SFileEntry::lpszFilePath 43  
SFileEntry::nAddress 43  
SFileEntry::nAttrib 43  
SFileEntry::nFileSize 43  
SFileEntry::pUserParam 43  
SFileTimeEx 44  
SFileTimeEx structure 44  
    about SFileTimeEx structure 44  
    CreationDateTime 46  
    LastAccessDateTime 46  
                ModificationDateTime 46  
                SFileTimeEx 45  
                SFileTimeEx data members 46  
                SFileTimeEx members 45  
                UseCreationDateTime 46  
                UseCustomTimes 47  
                UseLastAccessDateTime 47  
                UseModificationDateTime 47  
                SFileTimeEx::CreationDateTime 46  
                SFileTimeEx::LastAccessDateTime 46  
                SFileTimeEx::ModificationDateTime 46  
                SFileTimeEx::SFileTimeEx 45  
                SFileTimeEx::UseCreationDateTime 46  
                SFileTimeEx::UseCustomTimes 47  
                SFileTimeEx::UseLastAccessDateTime 47  
                SFileTimeEx::UseModificationDateTime 47  
                SFileToAdd 47  
                SFileToAdd structure 47  
                    about SFileToAdd structure 47  
                    bVideoFile 49  
                    INSERT\_STRUCTURE\_PADDING 50  
                    lpszDestinationPath 49  
                    lpszFileName 49  
                    lpszSourceFilePath 49  
                    nSavePath 49  
                    SFileToAdd 48  
                    SFileToAdd data members 48  
                    SFileToAdd members 48  
                    SFileToAdd methods 50  
                    SFileToAdd::bVideoFile 49  
                    SFileToAdd::INSERT\_STRUCTURE\_PADDING 50  
                    SFileToAdd::lpszDestinationPath 49  
                    SFileToAdd::lpszFileName 49  
                    SFileToAdd::lpszSourceFilePath 49  
                    SFileToAdd::nSavePath 49  
                    SFileToAdd::SFileToAdd 48  
                    SFileToAddEx 50  
                    SFileToAddEx structure 50  
                            about SFileToAddEx structure 50  
                            INSERT\_STRUCTURE\_PADDING 53  
                            lpszDestinationPath 51  
                            lpszFileName 52

IpszSourceFilePath	52	SFileToRename::IpszSourcePath	56
nFileOffset	52	SFileToRename::SFileToRename	55
nSavePath	52	SISOInfoEx	56
nSegmentSize	52	SISOInfoEx structure	56
SFileToAddEx	51	about SISOInfoEx structure	56
SFileToAddEx data members	51	INSERT_STRUCTURE_PADDING	64
SFileToAddEx members	50	ISOAbstractFileIdentifier	60
SFileToAddEx methods	52	ISOAddSuffix	60
SFileToAddEx::INSERT_STRUCTURE_PADDING	53	ISOAllowLongISO9660Names	60
SFileToAddEx::IpszDestinationPath	51	ISOAllowLongJolietNames	60
SFileToAddEx::IpszFileName	52	ISOAllowLowercaseNames	60
SFileToAddEx::IpszSourceFilePath	52	ISOAllowManyDirectories	61
SFileToAddEx::nFileOffset	52	ISOApplicationIdentifier	61
SFileToAddEx::nSavePath	52	ISOBiblioidentifier	61
SFileToAddEx::nSegmentSize	52	ISOCopyrightFileIdentifier	61
SFileToAddEx::SFileToAddEx	51	ISOCreationDateTime	61
SFileToRemove	53	ISODataPreparerIdentifier	61
SFileToRemove structure	53	ISOEffectiveDateTime	62
about SFileToRemove structure	53	ISOExpirationDateTime	62
INSERT_STRUCTURE_PADDING	54	ISOLevel	62
IpszDestinationPath	54	ISOModificationDateTime	62
IpszFile	54	ISOPublisherIdentifier	62
SFileToRemove	53	ISOSetIdentifier	62
SFileToRemove data members	54	ISOSystemIdentifier	63
SFileToRemove members	53	ISOUseCreationDateTime	63
SFileToRemove methods	54	ISOUseEffectiveDateTime	63
SFileToRemove::INSERT_STRUCTURE_PADDING	54	ISOUseExpirationDateTime	63
SFileToRemove::IpszDestinationPath	54	ISOUseModificationDateTime	63
SFileToRemove::IpszFile	54	SISOInfoEx	58
SFileToRemove::SFileToRemove	53	SISOInfoEx data members	58
SFileToRename	54	SISOInfoEx members	57
SFileToRename structure	54	SISOInfoEx methods	64
about SFileToRename structure	54	SISOInfoEx::INSERT_STRUCTURE_PADDING	64
INSERT_STRUCTURE_PADDING	56	SISOInfoEx::ISOAbstractFileIdentifier	60
IpszDestinationPath	55	SISOInfoEx::ISOAddSuffix	60
IpszSourcePath	56	SISOInfoEx::ISOAllowLongISO9660Names	60
SFileToRename	55	SISOInfoEx::ISOAllowLongJolietNames	60
SFileToRename data members	55	SISOInfoEx::ISOAllowLowercaseNames	60
SFileToRename members	55	SISOInfoEx::ISOAllowManyDirectories	61
SFileToRename methods	56	SISOInfoEx::ISOApplicationIdentifier	61
SFileToRename::INSERT_STRUCTURE_PADDING	56	SISOInfoEx::ISOBiblioidentifier	61
SFileToRename::IpszDestinationPath	55	SISOInfoEx::ISOCopyrightFileIdentifier	61

---

SISOInfoEx::ISOCreationDateTime 61  
SISOInfoEx::ISODataPreparerIdentifier 61  
SISOInfoEx::ISOEffectiveDateTime 62  
SISOInfoEx::ISOExpirationDateTime 62  
SISOInfoEx::ISOLevel 62  
SISOInfoEx::ISOModificationDateTime 62  
SISOInfoEx::ISOPublisherIdentifier 62  
SISOInfoEx::ISOSetIdentifier 62  
SISOInfoEx::ISOSystemIdentifier 63  
SISOInfoEx::ISOUseCreationDateTime 63  
SISOInfoEx::ISOUseEffectiveDateTime 63  
SISOInfoEx::ISOUseExpirationDateTime 63  
SISOInfoEx::ISOUseModificationDateTime 63  
SISOInfoEx::SISOInfoEx 58  
SISOVolumeInfo 64  
SISOVolumeInfo structure 64  
    about SISOVolumeInfo structure 64  
    chVolumeLabel 65  
    INSERT\_STRUCTURE\_PADDING 67  
    nPathTableAddress 65  
    nPathTableSize 66  
    nRootAddress 66  
    nVolumeDescriptorAddress 66  
    nVolumeSize 66  
    sInfoEx 66  
    SISOVolumeInfo 65  
    SISOVolumeInfo data members 65  
    SISOVolumeInfo members 64  
    SISOVolumeInfo methods 67  
    tRootDateTime 66  
SISOVolumeInfo::chVolumeLabel 65  
SISOVolumeInfo::INSERT\_STRUCTURE\_PADDING 67  
SISOVolumeInfo::nPathTableAddress 65  
SISOVolumeInfo::nPathTableSize 66  
SISOVolumeInfo::nRootAddress 66  
SISOVolumeInfo::nVolumeDescriptorAddress 66  
SISOVolumeInfo::nVolumeSize 66  
SISOVolumeInfo::sInfoEx 66  
SISOVolumeInfo::SISOVolumeInfo 65  
SISOVolumeInfo::tRootDateTime 66  
SMediumInfo 67  
SMediumInfo structure 67  
    about SMediumInfo structure 67  
    chMediumType 71  
    chUPCEANCode 71  
    chVendorID 71  
    dMediumFreeSize 72  
    dMediumSize 72  
    dMediumUsedSize 72  
    fMaxWriteSpeed 72  
    INSERT\_STRUCTURE\_PADDING 75  
    nExtendedMediumType 72  
    nFirstSession 73  
    nFirstTrack 73  
    nLastSession 73  
    nLastSessionStatus 73  
    nLastTrack 73  
    nMediumStatus 73  
    SMediumInfo 69  
    SMediumInfo data members 69  
    SMediumInfo members 67  
    SMediumInfo methods 75  
    wMediumTypeCode 74  
    SMediumInfo::chMediumType 71  
    SMediumInfo::chUPCEANCode 71  
    SMediumInfo::chVendorID 71  
    SMediumInfo::dMediumFreeSize 72  
    SMediumInfo::dMediumSize 72  
    SMediumInfo::dMediumUsedSize 72  
    SMediumInfo::fMaxWriteSpeed 72  
    SMediumInfo::INSERT\_STRUCTURE\_PADDING 75  
    SMediumInfo::nExtendedMediumType 72  
    SMediumInfo::nFirstSession 73  
    SMediumInfo::nFirstTrack 73  
    SMediumInfo::nLastSession 73  
    SMediumInfo::nLastSessionStatus 73  
    SMediumInfo::nLastTrack 73  
    SMediumInfo::nMediumStatus 73  
    SMediumInfo::SMediumInfo 69  
    SMediumInfo::wMediumTypeCode 74  
    SOptions 75  
    SOptions structure 75  
        about SOptions structure 75  
        bAutoErase 79

bBootable 79  
bEjectAfterBurn 79  
bFinalize 79  
bJolietFileSystem 79  
bPadDataTracks 80  
bPerformOPC 80  
bRockRidgeFileSystem 80  
bTestBurn 80  
bUnderrunProtection 80  
bVerifyAfterBurn 81  
chBootImage 81  
chVolumeLabel 81  
Empty 82  
INSERT\_STRUCTURE\_PADDING 82  
nCacheSize 81  
nCopies 81  
nWriteMethod 81  
SOptions 77  
SOptions data members 78  
SOptions members 76  
SOptions methods 82  
SOptions::bAutoErase 79  
SOptions::bBootable 79  
SOptions::bEjectAfterBurn 79  
SOptions::bFinalize 79  
SOptions::bJolietFileSystem 79  
SOptions::bPadDataTracks 80  
SOptions::bPerformOPC 80  
SOptions::bRockRidgeFileSystem 80  
SOptions::bTestBurn 80  
SOptions::bUnderrunProtection 80  
SOptions::bVerifyAfterBurn 81  
SOptions::chBootImage 81  
SOptions::chVolumeLabel 81  
SOptions::Empty 82  
SOptions::INSERT\_STRUCTURE\_PADDING 82  
SOptions::nCacheSize 81  
SOptions::nCopies 81  
SOptions::nWriteMethod 81  
SOptions::SOptions 77  
SPrintf 514  
SPrintf macro 514  
SRAWTrackToAdd 82  
SRAWTrackToAdd structure 82  
    about SRAWTrackToAdd structure 82  
    INSERT\_STRUCTURE\_PADDING 86  
    nDataTypeMask 84  
    nFormat 84  
    nIgnoreDataMask 85  
    nIndex 85  
    nLength 85  
    nNumber 85  
    nOffset 85  
    nStartAddress 85  
    SRAWTrackToAdd 83  
    SRAWTrackToAdd data members 83  
    SRAWTrackToAdd members 83  
    SRAWTrackToAdd methods 86  
    SRAWTrackToAdd::INSERT\_STRUCTURE\_PADDING 86  
    SRAWTrackToAdd::nDataTypeMask 84  
    SRAWTrackToAdd::nFormat 84  
    SRAWTrackToAdd::nIgnoreDataMask 85  
    SRAWTrackToAdd:: nIndex 85  
    SRAWTrackToAdd::nLength 85  
    SRAWTrackToAdd::nNumber 85  
    SRAWTrackToAdd::nOffset 85  
    SRAWTrackToAdd::nStartAddress 85  
    SRAWTrackToAdd::SRAWTrackToAdd 83  
    SReadErrorCorrectionParams 86  
    SReadErrorCorrectionParams structure 86  
        about SReadErrorCorrectionParams structure 86  
        bBlankBadSectors 87  
        bErrorCorrection 87  
        INSERT\_STRUCTURE\_PADDING 88  
        nHardwareRetryCount 87  
        nSoftwareRetryCount 88  
        SReadErrorCorrectionParams 87  
        SReadErrorCorrectionParams data members 87  
        SReadErrorCorrectionParams members 86  
        SReadErrorCorrectionParams methods 88  
    SReadErrorCorrectionParams::bBlankBadSectors 87  
    SReadErrorCorrectionParams::bErrorCorrection 87  
    SReadErrorCorrectionParams::INSERT\_STRUCTURE\_PADDING 88

SReadErrorCorrectionParams::nHardwareRetryCount	87	chComment	93
SReadErrorCorrectionParams::nSoftwareRetryCount	88	chTitle	94
SReadErrorCorrectionParams::SReadErrorCorrectionParams	87	Empty	94
SSessionInfo	88	STags	93
SSessionInfo structure	88	STags data members	93
about SSessionInfo structure	88	STags members	92
bLastSession	89	STags methods	94
dSessionSize	89	STags::chAlbum	93
dStartLBA	90	STags::chArtist	93
INSERT_STRUCTURE_PADDING	90	STags::chComment	93
nFirstTrack	90	STags::chTitle	94
nLastTrack	90	STags::Empty	94
SSessionInfo	89	STags::STags	93
SSessionInfo data members	89	StartVerifyEvent	243
SSessionInfo members	88	StartVerifyEvent type	243
SSessionInfo methods	90	StopMpegAction	229
SSessionInfo::bLastSession	89	StopMpegAction function	229
SSessionInfo::dSessionSize	89	STrackInfo	94
SSessionInfo::dStartLBA	90	STrackInfo structure	94
SSessionInfo::INSERT_STRUCTURE_PADDING	90	about STrackInfo structure	94
SSessionInfo::nFirstTrack	90	INSERT_STRUCTURE_PADDING	97
SSessionInfo::nLastTrack	90	nFileSystem	96
SSessionInfo::SSessionInfo	89	nFormat	96
SSpeed	90	nSessionNumber	96
SSpeed structure	90	nSize	96
about SSpeed structure	90	nStartLBA	96
fSpeed	91	nTrackNumber	97
INSERT_STRUCTURE_PADDING	92	STrackInfo	95
nSpeedInKBPerSec	91	STrackInfo data members	95
SSpeed	91	STrackInfo members	94
SSpeed data members	91	STrackInfo methods	97
SSpeed members	91	STrackInfo::INSERT_STRUCTURE_PADDING	97
SSpeed methods	92	STrackInfo::nFileSystem	96
SSpeed::fSpeed	91	STrackInfo::nFormat	96
SSpeed::INSERT_STRUCTURE_PADDING	92	STrackInfo::nSessionNumber	96
SSpeed::nSpeedInKBPerSec	91	STrackInfo::nSize	96
SSpeed::SSpeed	91	STrackInfo::nStartLBA	96
STags	92	STrackInfo::nTrackNumber	97
STags structure	92	STrackInfo::STrackInfo	95
about STags structure	92	StrCmp	514
chAlbum	93	StrCmp macro	514
chArtist	93	StrCpy	514

StrCpy macro	514	nDirCount	102
StrLen	514	nFileCount	102
StrLen macro	514	nFSDAddress	103
StrnSet	515	nLVDAddress	103
StrnSet macro	515	nMetadataAddress	103
Structs, Records, Enums	230	nMVDSAddress	103
STRUCTURE_PADDING_BIG	515	nPartitionAddress	103
STRUCTURE_PADDING_BIG macro	515	nPartitionLength	103
STRUCTURE_PADDING_HUGE	515	nPartitionType	103
STRUCTURE_PADDING_HUGE macro	515	nPVDAddress	104
STRUCTURE_PADDING_NORMAL	515	nRootAddress	104
STRUCTURE_PADDING_NORMAL macro	515	nRootFEAddress	104
STRUCTURE_PADDING_SMALL	516	nRVDSAddress	104
STRUCTURE_PADDING_SMALL macro	516	nSparingAddress	104
STRUCTURE_PADDING_STRING_DATA	516	nVATAddress	104
STRUCTURE_PADDING_STRING_DATA macro	516	nVersion	105
SUDFOptions	97	SUDFVolumeInfo	101
SUDFOptions structure	97	SUDFVolumeInfo data members	101
about SUDFOptions structure	97	SUDFVolumeInfo members	100
AvchdDisc	98	SUDFVolumeInfo methods	105
INSERT_STRUCTURE_PADDING	100	tRecordingDateTime	105
PartitionType	99	SUDFVolumeInfo::chPreparer	102
SUDFOptions	98	SUDFVolumeInfo::chVolumeLabel	102
SUDFOptions data members	98	SUDFVolumeInfo::INSERT_STRUCTURE_PADDING	105
SUDFOptions members	97	SUDFVolumeInfo::nDirCount	102
SUDFOptions methods	99	SUDFVolumeInfo::nFileCount	102
Version	99	SUDFVolumeInfo::nFSDAddress	103
WriteFileStreams	99	SUDFVolumeInfo::nLVDAddress	103
SUDFOptions::AvchdDisc	98	SUDFVolumeInfo::nMetadataAddress	103
SUDFOptions::INSERT_STRUCTURE_PADDING	100	SUDFVolumeInfo::nMVDSAddress	103
SUDFOptions::PartitionType	99	SUDFVolumeInfo::nPartitionAddress	103
SUDFOptions::SUDFOptions	98	SUDFVolumeInfo::nPartitionLength	103
SUDFOptions::Version	99	SUDFVolumeInfo::nPartitionType	103
SUDFOptions::WriteFileStreams	99	SUDFVolumeInfo::nPVDAddress	104
SUDFOptionsEx	232	SUDFVolumeInfo::nRootAddress	104
SUDFOptionsEx structure	232	SUDFVolumeInfo::nRootFEAddress	104
SUDFVolumeInfo	100	SUDFVolumeInfo::nRVDSAddress	104
SUDFVolumeInfo structure	100	SUDFVolumeInfo::nSparingAddress	104
about SUDFVolumeInfo structure	100	SUDFVolumeInfo::nVATAddress	104
chPreparer	102	SUDFVolumeInfo::nVersion	105
chVolumeLabel	102	SUDFVolumeInfo::SUDFVolumeInfo	101
INSERT_STRUCTURE_PADDING	105	SUDFVolumeInfo::tRecordingDateTime	105

SVideoFormat 105  
SVideoFormat structure 105  
    about SVideoFormat structure 105  
    bUseable 107  
    fFPS 107  
    INSERT\_STRUCTURE\_PADDING 108  
    nAspectRatio 107  
    nBitRate 107  
    nHeight 108  
    nPlaytime 108  
    nWidth 108  
    SVideoFormat 106  
    SVideoFormat data members 106  
    SVideoFormat members 106  
    SVideoFormat methods 108  
SVideoFormat::bUseable 107  
SVideoFormat::fFPS 107  
SVideoFormat::INSERT\_STRUCTURE\_PADDING 108  
SVideoFormat::nAspectRatio 107  
SVideoFormat::nBitRate 107  
SVideoFormat::nHeight 108  
SVideoFormat::nPlaytime 108  
SVideoFormat::nWidth 108  
SVideoFormat::SVideoFormat 106

## T

TagsFromNetworkDialog 229  
TagsFromNetworkDialog function 229  
Types 234

## U

uint16 243  
uint16 type 243  
uint32 244  
uint32 type 244  
uint8 244  
uint8 type 244

## V

VerifyDoneEvent 244  
VerifyDoneEvent type 244  
VerifyErrorEvent 245