

NAS VAAI Development Kit - ESX Server 7.0.1
API Reference vmkapi_2_7_0_0

Generated by Doxygen 1.5.8

Fri Sep 4 11:40:47 2020

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	VixDiskLibBlock Struct Reference	5
3.1.1	Field Documentation	5
3.1.1.1	offset	5
3.1.1.2	length	5
3.2	VixDiskLibBlockList Struct Reference	6
3.2.1	Field Documentation	6
3.2.1.1	numBlocks	6
3.2.1.2	blocks	6
3.3	VixDiskLibConnectParams Struct Reference	7
3.3.1	Detailed Description	7
3.3.2	Field Documentation	8
3.3.2.1	vmxSpec	8
3.3.2.2	serverName	8
3.3.2.3	thumbPrint	8
3.3.2.4	privateUse	8
3.3.2.5	credType	8
3.3.2.6	creds	8
3.3.2.7	port	8
3.3.2.8	nfcHostPort	8
3.3.2.9	vimApiVer	8
3.3.2.10	reserved	8
3.3.2.11	state	8

3.3.2.12	vStorageObjSpec	8
3.3.2.13	dsSpec	8
3.3.2.14	spec	8
3.3.2.15	specType	8
3.4	VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds Union Reference	9
3.4.1	Field Documentation	9
3.4.1.1	uid	9
3.4.1.2	sessionId	9
3.4.1.3	ticketId	9
3.5	VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds Struct Reference	10
3.5.1	Field Documentation	10
3.5.1.1	cookie	10
3.5.1.2	userName	10
3.5.1.3	key	10
3.6	VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds Struct Reference	11
3.6.1	Field Documentation	11
3.6.1.1	userName	11
3.6.1.2	password	11
3.7	VixDiskLibCreateParams Struct Reference	12
3.7.1	Field Documentation	12
3.7.1.1	diskType	12
3.7.1.2	adapterType	12
3.7.1.3	hwVersion	12
3.7.1.4	capacity	12
3.7.1.5	logicalSectorSize	12
3.7.1.6	physicalSectorSize	12
3.8	VixDiskLibDatastoreSpec Struct Reference	13
3.8.1	Field Documentation	13
3.8.1.1	datastoreMoRef	13
3.9	VixDiskLibGeometry Struct Reference	14
3.9.1	Field Documentation	14
3.9.1.1	cylinders	14
3.9.1.2	heads	14
3.9.1.3	sectors	14
3.10	VixDiskLibInfo Struct Reference	15

3.10.1	Field Documentation	15
3.10.1.1	biosGeo	15
3.10.1.2	physGeo	15
3.10.1.3	capacity	15
3.10.1.4	adapterType	15
3.10.1.5	numLinks	15
3.10.1.6	parentFileNameHint	15
3.10.1.7	uuid	15
3.10.1.8	logicalSectorSize	15
3.10.1.9	physicalSectorSize	15
3.11	VixDiskLibNasPlugin Struct Reference	16
3.11.1	Detailed Description	16
3.11.2	Field Documentation	16
3.11.2.1	diskLibPlugin	16
3.11.2.2	StartSession	16
3.11.2.3	EndSession	16
3.11.2.4	ExecPrimitive	16
3.11.2.5	SupportStatus	16
3.12	VixDiskLibNasPluginCloneFileParams Struct Reference	17
3.12.1	Field Documentation	17
3.12.1.1	common	17
3.12.1.2	srcFileName	17
3.12.1.3	dstFileName	17
3.12.1.4	cloneFlags	17
3.12.1.5	srcDataStoreInfo	17
3.13	VixDiskLibNasPluginCommonParams Struct Reference	18
3.13.1	Field Documentation	18
3.13.1.1	primitiveID	18
3.13.1.2	progressRecord	18
3.13.1.3	result	18
3.14	VixDiskLibNasPluginDataStoreParams Struct Reference	19
3.14.1	Field Documentation	19
3.14.1.1	fsType	19
3.14.1.2	fsVersion	19
3.14.1.3	remoteIP	19
3.14.1.4	remoteMountPoint	19

3.14.1.5	localMountPoint	19
3.15	VixDiskLibNasPluginProgressRecord Struct Reference	20
3.15.1	Field Documentation	20
3.15.1.1	private	20
3.15.1.2	updateBytes	20
3.15.1.3	progressBytes	20
3.15.1.4	callback	20
3.16	VixDiskLibNasPluginResultCommon Struct Reference	21
3.16.1	Field Documentation	21
3.16.1.1	status	21
3.17	VixDiskLibNasPluginResvSpaceParams Struct Reference	22
3.17.1	Field Documentation	22
3.17.1.1	common	22
3.17.1.2	fileName	22
3.18	VixDiskLibNasPluginSessionParams Struct Reference	23
3.18.1	Field Documentation	23
3.18.1.1	timeoutMS	23
3.19	VixDiskLibNasPluginStatXParams Struct Reference	24
3.19.1	Field Documentation	24
3.19.1.1	common	24
3.19.1.2	fileName	24
3.20	VixDiskLibNasPluginStatXResult Struct Reference	25
3.20.1	Field Documentation	25
3.20.1.1	common	25
3.20.1.2	totalBytes	25
3.20.1.3	allocatedBytes	25
3.20.1.4	uniqueBytes	25
3.20.1.5	allocType	25
3.21	VixDiskLibPlugin Struct Reference	26
3.21.1	Detailed Description	26
3.21.2	Field Documentation	26
3.21.2.1	majorVersion	26
3.21.2.2	minorVersion	26
3.21.2.3	type	26
3.21.2.4	name	26
3.21.2.5	Init	26

3.21.2.6	Done	26
3.22	VixDiskLibSpec Union Reference	27
3.22.1	Field Documentation	27
3.22.1.1	vmxSpec	27
3.22.1.2	vStorageObjSpec	27
3.22.1.3	dsSpec	27
3.23	VixDiskLibVStorageObjectSpec Struct Reference	28
3.23.1	Field Documentation	28
3.23.1.1	id	28
3.23.1.2	datastoreMoRef	28
3.23.1.3	ssId	28
3.24	VMPoint Struct Reference	29
3.24.1	Field Documentation	29
3.24.1.1	x	29
3.24.1.2	y	29
3.25	VMRect Struct Reference	30
3.25.1	Field Documentation	30
3.25.1.1	left	30
3.25.1.2	top	30
3.25.1.3	right	30
3.25.1.4	bottom	30
4	File Documentation	31
4.1	distribute_vixDiskLib.h File Reference	31
4.1.1	Define Documentation	36
4.1.1.1	VIXDISKLIB_SECTOR_SIZE	36
4.1.1.2	VIXDISKLIB_HWVERSION_WORKSTATION_4	36
4.1.1.3	VIXDISKLIB_HWVERSION_WORKSTATION_5	36
4.1.1.4	VIXDISKLIB_HWVERSION_WORKSTATION_6	36
4.1.1.5	VIXDISKLIB_HWVERSION_ESX30	36
4.1.1.6	VIXDISKLIB_HWVERSION_ESX4X	36
4.1.1.7	VIXDISKLIB_HWVERSION_ESX50	36
4.1.1.8	VIXDISKLIB_HWVERSION_ESX51	36
4.1.1.9	VIXDISKLIB_HWVERSION_ESX55	36
4.1.1.10	VIXDISKLIB_HWVERSION_ESX60	36
4.1.1.11	VIXDISKLIB_HWVERSION_ESX65	36
4.1.1.12	VIXDISKLIB_HWVERSION_CURRENT	36

4.1.1.13	VIXDISKLIB_MIN_CHUNK_SIZE	36
4.1.1.14	VIXDISKLIB_MAX_CHUNK_SIZE	36
4.1.1.15	VIXDISKLIB_MAX_CHUNK_NUMBER	36
4.1.1.16	VIXDISKLIB_FLAG_OPEN_UNBUFFERED	36
4.1.1.17	VIXDISKLIB_FLAG_OPEN_SINGLE_LINK	36
4.1.1.18	VIXDISKLIB_FLAG_OPEN_READ_ONLY	36
4.1.1.19	VIXDISKLIB_FLAG_OPEN_COMPRESSION_ZLIB	36
4.1.1.20	VIXDISKLIB_FLAG_OPEN_COMPRESSION_FASTLZ	36
4.1.1.21	VIXDISKLIB_FLAG_OPEN_COMPRESSION_SKIPZ	36
4.1.1.22	VIXDISKLIB_FLAG_OPEN_COMPRESSION_MASK	36
4.1.2	Typedef Documentation	36
4.1.2.1	VixDiskLibSectorType	36
4.1.2.2	VixDiskLibConnectParamsState	36
4.1.2.3	VixDiskLibHandleStruct	36
4.1.2.4	VixDiskLibHandle	36
4.1.2.5	VixDiskLibConnection	36
4.1.2.6	VixDiskLibGenericLogFunc	36
4.1.2.7	VixDiskLibGenericLogVFunc	36
4.1.2.8	VixDiskLibProgressFunc	36
4.1.2.9	VixDiskLibCompletionCB	36
4.1.3	Enumeration Type Documentation	36
4.1.3.1	VixDiskLibDiskType	36
4.1.3.2	VixDiskLibAdapterType	37
4.1.3.3	VixDiskLibCredType	37
4.1.3.4	VixDiskLibSpecType	37
4.1.4	Function Documentation	37
4.1.4.1	VixDiskLib_InitEx	37
4.1.4.2	VixDiskLib_Init	38
4.1.4.3	VixDiskLib_Exit	38
4.1.4.4	VixDiskLib_ListTransportModes	38
4.1.4.5	VixDiskLib_Cleanup	39
4.1.4.6	VixDiskLib_Connect	39
4.1.4.7	VixDiskLib_PrepareForAccess	39
4.1.4.8	VixDiskLib_ConnectEx	40
4.1.4.9	VixDiskLib_Disconnect	40
4.1.4.10	VixDiskLib_EndAccess	40

4.1.4.11	VixDiskLib_Create	41
4.1.4.12	VixDiskLib_CreateChild	41
4.1.4.13	VixDiskLib_Open	42
4.1.4.14	VixDiskLib_QueryAllocatedBlocks	42
4.1.4.15	VixDiskLib_FreeBlockList	42
4.1.4.16	VixDiskLib_GetInfo	42
4.1.4.17	VixDiskLib_FreeInfo	43
4.1.4.18	VixDiskLib_GetTransportMode	43
4.1.4.19	VixDiskLib_Close	43
4.1.4.20	VixDiskLib_Read	43
4.1.4.21	VixDiskLib_ReadAsync	44
4.1.4.22	VixDiskLib_Write	44
4.1.4.23	VixDiskLib_WriteAsync	44
4.1.4.24	VixDiskLib_Flush	45
4.1.4.25	VixDiskLib_Wait	45
4.1.4.26	VixDiskLib_ReadMetadata	45
4.1.4.27	VixDiskLib_WriteMetadata	45
4.1.4.28	VixDiskLib_GetMetadataKeys	46
4.1.4.29	VixDiskLib_Unlink	46
4.1.4.30	VixDiskLib_Grow	46
4.1.4.31	VixDiskLib_Shrink	47
4.1.4.32	VixDiskLib_Defragment	47
4.1.4.33	VixDiskLib_Rename	47
4.1.4.34	VixDiskLib_Clone	48
4.1.4.35	VixDiskLib_GetErrorText	48
4.1.4.36	VixDiskLib_FreeErrorText	48
4.1.4.37	VixDiskLib_IsAttachPossible	49
4.1.4.38	VixDiskLib_Attach	49
4.1.4.39	VixDiskLib_SpaceNeededForClone	49
4.1.4.40	VixDiskLib_CheckRepair	49
4.1.4.41	VixDiskLib_GetConnectParams	50
4.1.4.42	VixDiskLib_FreeConnectParams	50
4.1.4.43	VixDiskLib_AllocateConnectParams	50
4.2	distribute_vixDiskLibNasPlugin.h File Reference	51
4.2.1	Define Documentation	52
4.2.1.1	VIXDISKLIB_NASPLUGIN_MAJOR_VERSION	52

4.2.1.2	VIXDISKLIB_NASPLUGIN_MINOR_VERSION	52
4.2.1.3	VIXDISKLIB_NASPLUGIN_FSTYPE_NFS	52
4.2.1.4	VIXDISKLIB_NASPLUGIN_FSTYPE_NFS41	52
4.2.1.5	VIXDISKLIB_NASPLUGIN_FSTYPE_VMFS	52
4.2.1.6	VIXDISKLIB_NASPLUGIN_FSTYPE_VMFSL	52
4.2.1.7	VIXDISKLIB_NASPLUGIN_INVALID_SESSION_ID	52
4.2.2	Typedef Documentation	52
4.2.2.1	VixDiskLibNasPluginSessionID	52
4.2.2.2	VixDiskLibNasPluginPeriodicCallback	52
4.2.2.3	VixDiskLibNasPluginStartSession	52
4.2.2.4	VixDiskLibNasPluginEndSession	53
4.2.2.5	VixDiskLibNasPluginSupportStatus	53
4.2.2.6	VixDiskLibNasPluginExecutePrimitive	53
4.2.3	Enumeration Type Documentation	54
4.2.3.1	VixDiskLibNasPluginPrimitiveID	54
4.2.3.2	VixDiskLibNasPluginAllocType	54
4.2.3.3	VixDiskLibNasPluginCloneFileFlags	55
4.3	distributed_vixDiskLibPlugin.h File Reference	56
4.3.1	Define Documentation	56
4.3.1.1	VIXDISKLIB_PLUGIN_MAJOR_VERSION	56
4.3.1.2	VIXDISKLIB_PLUGIN_MINOR_VERSION	56
4.3.2	Typedef Documentation	56
4.3.2.1	VixDiskLibPluginInit	56
4.3.2.2	VixDiskLibPluginDone	57
4.3.3	Enumeration Type Documentation	57
4.3.3.1	VixDiskLibPluginType	57
4.3.4	Variable Documentation	57
4.3.4.1	VixDiskLibPlugin_EntryPoint	57
4.4	public_vm_basic_types.h File Reference	58
4.4.1	Define Documentation	63
4.4.1.1	INCLUDE_ALLOW_USERLEVEL	63
4.4.1.2	INCLUDE_ALLOW_MODULE	63
4.4.1.3	INCLUDE_ALLOW_VMMON	63
4.4.1.4	INCLUDE_ALLOW_VMKERNEL	63
4.4.1.5	INCLUDE_ALLOW_VMKDRIVERS	63
4.4.1.6	INCLUDE_ALLOW_VMK_MODULE	63

4.4.1.7	INCLUDE_ALLOW_DISTRIBUTE	63
4.4.1.8	INCLUDE_ALLOW_VMCORE	63
4.4.1.9	vm_x86_64	63
4.4.1.10	vm_arm_64	63
4.4.1.11	vm_64bit	63
4.4.1.12	_XTYPEDEF_BOOL	63
4.4.1.13	FALSE	63
4.4.1.14	TRUE	63
4.4.1.15	IS_BOOL	63
4.4.1.16	CONST3264	63
4.4.1.17	CONST3264U	63
4.4.1.18	MIN_INT8	63
4.4.1.19	MAX_INT8	63
4.4.1.20	MIN_UINT8	63
4.4.1.21	MAX_UINT8	63
4.4.1.22	MIN_INT16	63
4.4.1.23	MAX_INT16	63
4.4.1.24	MIN_UINT16	63
4.4.1.25	MAX_UINT16	63
4.4.1.26	MIN_INT32	63
4.4.1.27	MAX_INT32	63
4.4.1.28	MIN_UINT32	63
4.4.1.29	MAX_UINT32	63
4.4.1.30	MIN_INT64	63
4.4.1.31	MAX_INT64	63
4.4.1.32	MIN_UINT64	63
4.4.1.33	MAX_UINT64	63
4.4.1.34	AsPercent	63
4.4.1.35	UINT64_2_BPN	63
4.4.1.36	BPN_2_UINT64	63
4.4.1.37	INVALID_WORLD_ID	63
4.4.1.38	INVALID_CARTEL_ID	63
4.4.1.39	INVALID_SESSION_ID	63
4.4.1.40	INVALID_CARTELGROUP_ID	63
4.4.1.41	INVALID_WORLDLET_ID	63
4.4.1.42	LA_2_LPN	63

4.4.1.43	LPN_2_LA	63
4.4.1.44	LAST_LPN	63
4.4.1.45	LAST_LPN32	63
4.4.1.46	LAST_LPN64	63
4.4.1.47	LPN_MASK	63
4.4.1.48	LPN_MASK32	63
4.4.1.49	LPN_MASK64	63
4.4.1.50	MAX_PPN_BITS	63
4.4.1.51	MAX_PPN	63
4.4.1.52	INVALID_PPN	63
4.4.1.53	INVALID_PPN32	63
4.4.1.54	APIC_INVALID_PPN	63
4.4.1.55	INVALID_BPN	63
4.4.1.56	MPN38_MASK	63
4.4.1.57	RESERVED_MPN	63
4.4.1.58	INVALID_MPN	63
4.4.1.59	MEMREF_MPN	63
4.4.1.60	RELEASED_MPN	63
4.4.1.61	MAX_MPN	63
4.4.1.62	INVALID_IOPN	63
4.4.1.63	MAX_IOPN	63
4.4.1.64	INVALID_LPN	63
4.4.1.65	INVALID_VPN	63
4.4.1.66	INVALID_LPN64	63
4.4.1.67	INVALID_PAGENUM	63
4.4.1.68	INVALID_PAGENUM32	63
4.4.1.69	FMTLA	63
4.4.1.70	FMTVA	63
4.4.1.71	FMTVPN	63
4.4.1.72	EXTERN	63
4.4.1.73	CONST	63
4.4.1.74	INLINE	63
4.4.1.75	VMX86_EXTERN_DATA	63
4.4.1.76	INLINE_ALWAYS	63
4.4.1.77	INLINE_SINGLE_CALLER	63
4.4.1.78	SIDE_EFFECT_FREE	63

4.4.1.79	CONST_FUNCTION	63
4.4.1.80	NORETURN	63
4.4.1.81	HOT	63
4.4.1.82	COLD	63
4.4.1.83	LIKELY	63
4.4.1.84	UNLIKELY	63
4.4.1.85	PRINTF_DECL	63
4.4.1.86	SCANF_DECL	63
4.4.1.87	UNUSED_PARAM	63
4.4.1.88	UNUSED_TYPE	63
4.4.1.89	UNUSED_VARIABLE	63
4.4.1.90	MUST_CHECK_RETURN	63
4.4.1.91	ALIGNED	63
4.4.1.92	INFINITE_LOOP	63
4.4.1.93	FMTPID	63
4.4.1.94	FMTUID	63
4.4.1.95	FMTMODE	63
4.4.2	Typedef Documentation	63
4.4.2.1	uint64	63
4.4.2.2	int64	63
4.4.2.3	uint32	63
4.4.2.4	int32	63
4.4.2.5	uint16	63
4.4.2.6	int16	63
4.4.2.7	uint8	63
4.4.2.8	int8	63
4.4.2.9	Bool	63
4.4.2.10	VmTimeType	63
4.4.2.11	VmTimeRealClock	63
4.4.2.12	VmTimeVirtualClock	63
4.4.2.13	TCA	63
4.4.2.14	Percent	63
4.4.2.15	VA	63
4.4.2.16	VPN	63
4.4.2.17	PA	63
4.4.2.18	PPN	63

4.4.2.19	TPA	63
4.4.2.20	TPPN	63
4.4.2.21	PhysMemOff	63
4.4.2.22	PhysMemSize	63
4.4.2.23	BA	63
4.4.2.24	BPN	63
4.4.2.25	PageCnt	63
4.4.2.26	PageNum	63
4.4.2.27	MemHandle	63
4.4.2.28	IoHandle	63
4.4.2.29	World_ID	63
4.4.2.30	User_CartelID	63
4.4.2.31	User_SessionID	63
4.4.2.32	User_CartelGroupID	63
4.4.2.33	Worldlet_ID	63
4.4.2.34	Reg8	63
4.4.2.35	Reg16	63
4.4.2.36	Reg32	63
4.4.2.37	Reg64	63
4.4.2.38	UReg8	63
4.4.2.39	UReg16	63
4.4.2.40	UReg32	63
4.4.2.41	UReg64	63
4.4.2.42	MA	63
4.4.2.43	MPN32	63
4.4.2.44	SectorType	63
4.4.2.45	LA	63
4.4.2.46	LPN	63
4.4.2.47	VA32	63
4.4.2.48	VPN32	63
4.4.2.49	LA32	63
4.4.2.50	LPN32	63
4.4.2.51	PA32	63
4.4.2.52	PPN32	63
4.4.2.53	VA64	63
4.4.2.54	VPN64	63

4.4.2.55	LA64	63
4.4.2.56	LPN64	63
4.4.2.57	PA64	63
4.4.2.58	PPN64	63
4.4.2.59	TPPN64	63
4.4.2.60	MA64	63
4.4.2.61	MPN	63
4.4.2.62	IOA	63
4.4.2.63	IOPN	63
4.4.2.64	UserVA32	63
4.4.2.65	UserVA64	63
4.4.2.66	UserVAConst	63
4.4.2.67	UserVA32Const	63
4.4.2.68	UserVA64Const	63
4.4.2.69	UserVA	63
4.4.2.70	PollDevHandle	63
4.4.2.71	utf16_t	63
4.4.2.72	MX_Rank	63

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

VixDiskLibBlock	5
VixDiskLibBlockList	6
VixDiskLibConnectParams	7
VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds	9
VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds	10
VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds	11
VixDiskLibCreateParams	12
VixDiskLibDatastoreSpec	13
VixDiskLibGeometry	14
VixDiskLibInfo	15
VixDiskLibNasPlugin	16
VixDiskLibNasPluginCloneFileParams	17
VixDiskLibNasPluginCommonParams	18
VixDiskLibNasPluginDataStoreParams	19
VixDiskLibNasPluginProgressRecord	20
VixDiskLibNasPluginResultCommon	21
VixDiskLibNasPluginResvSpaceParams	22
VixDiskLibNasPluginSessionParams	23
VixDiskLibNasPluginStatXParams	24
VixDiskLibNasPluginStatXResult	25
VixDiskLibPlugin	26
VixDiskLibSpec	27
VixDiskLibVStorageObjectSpec	28
VMPoint	29
VMRect	30

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

distribute_vixDiskLib.h	31
distribute_vixDiskLibNasPlugin.h	51
distribute_vixDiskLibPlugin.h	56
public_vm_basic_types.h	58

Chapter 3

Data Structure Documentation

3.1 VixDiskLibBlock Struct Reference

Data Fields

- [VixDiskLibSectorType offset](#)
- [VixDiskLibSectorType length](#)

3.1.1 Field Documentation

3.1.1.1 VixDiskLibSectorType VixDiskLibBlock::offset

3.1.1.2 VixDiskLibSectorType VixDiskLibBlock::length

3.2 VixDiskLibBlockList Struct Reference

Data Fields

- [uint32 numBlocks](#)
- [VixDiskLibBlock blocks](#) [1]

3.2.1 Field Documentation

3.2.1.1 [uint32 VixDiskLibBlockList::numBlocks](#)

3.2.1.2 [VixDiskLibBlock VixDiskLibBlockList::blocks\[1\]](#)

3.3 VixDiskLibConnectParams Struct Reference

Data Structures

- union [VixDiskLibCreds](#)

Data Fields

- char * [vmxSpec](#)
- char * [serverName](#)
- char * [thumbPrint](#)
- long [privateUse](#)
- [VixDiskLibCredType](#) [credType](#)
- union [VixDiskLibConnectParams::VixDiskLibCreds](#) [creds](#)
- [uint32](#) [port](#)
- [uint32](#) [nfcHostPort](#)
- char * [vimApiVer](#)
- char [reserved](#) [8]
- [VixDiskLibConnectParamsState](#) * [state](#)
- union {
 - [VixDiskLibVStorageObjectSpec](#) [vStorageObjSpec](#)
 - [VixDiskLibDatastoreSpec](#) [dsSpec](#)} [spec](#)
- [VixDiskLibSpecType](#) [specType](#)

3.3.1 Detailed Description

[VixDiskLibConnectParams](#) - Connection setup parameters.

[vmxSpec](#) is required for opening a virtual disk on a datastore through the Virtual Center or ESX server. [vmxSpec](#) is of the form: <[vmxPathName](#)>?dcPath=<[dcpath](#)>&dsName=<[dsname](#)> where [vmxPathName](#) is the fullpath for the VMX file, [dcpath](#) is the inventory path of the datacenter and [dsname](#) is the datastore name.

Inventory path for the datacenter can be read off the Virtual Center client's inventory tree.

Example VM spec: "MyVm/MyVm.vmx?dcPath=Path/to/MyDatacenter&dsName=storage1"

3.3.2 Field Documentation

3.3.2.1 char* VixDiskLibConnectParams::vmxSpec

3.3.2.2 char* VixDiskLibConnectParams::serverName

3.3.2.3 char* VixDiskLibConnectParams::thumbPrint

3.3.2.4 long VixDiskLibConnectParams::privateUse

3.3.2.5 VixDiskLibCredType VixDiskLibConnectParams::credType

3.3.2.6 union VixDiskLibConnectParams::VixDiskLibCreds VixDiskLibConnectParams::creds

3.3.2.7 uint32 VixDiskLibConnectParams::port

3.3.2.8 uint32 VixDiskLibConnectParams::nfcHostPort

3.3.2.9 char* VixDiskLibConnectParams::vimApiVer

3.3.2.10 char VixDiskLibConnectParams::reserved[8]

3.3.2.11 VixDiskLibConnectParamsState* VixDiskLibConnectParams::state

3.3.2.12 VixDiskLibVStorageObjectSpec VixDiskLibConnectParams::vStorageObjSpec

3.3.2.13 VixDiskLibDatastoreSpec VixDiskLibConnectParams::dsSpec

3.3.2.14 union { ... } VixDiskLibConnectParams::spec

3.3.2.15 VixDiskLibSpecType VixDiskLibConnectParams::specType

3.4 VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds Union Reference

Data Structures

- struct [VixDiskLibSessionIdCreds](#)
- struct [VixDiskLibUidPasswdCreds](#)

Data Fields

- struct [VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibUidPasswdCreds](#) uid
- struct [VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibSessionIdCreds](#) sessionId
- struct [VixDiskLibTicketIdCreds](#) * ticketId

3.4.1 Field Documentation

3.4.1.1 struct [VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibUidPasswdCreds](#)
[VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::uid](#)

3.4.1.2 struct [VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibSessionIdCreds](#)
[VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::sessionId](#)

3.4.1.3 struct [VixDiskLibTicketIdCreds](#)* [VixDiskLibConnect-](#)
[Params::VixDiskLibConnectParams::VixDiskLibCreds::ticketId](#)
[read]

3.5 VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds:: Struct Reference

Data Fields

- char * [cookie](#)
- char * [userName](#)
- char * [key](#)

3.5.1 Field Documentation

3.5.1.1 char* VixDiskLibConnect-Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixD

3.5.1.2 char* VixDiskLibConnect-Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixD

3.5.1.3 char* VixDiskLibConnect-Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixD

3.6 VixDiskLibConnect-

Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibU
Struct Reference

3.6 VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::¹¹ Struct Reference

Data Fields

- char * [userName](#)
- char * [password](#)

3.6.1 Field Documentation

3.6.1.1 char* VixDiskLibConnect-
Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixD

3.6.1.2 char* VixDiskLibConnect-
Params::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds::VixD

3.7 VixDiskLibCreateParams Struct Reference

Data Fields

- [VixDiskLibDiskType](#) diskType
- [VixDiskLibAdapterType](#) adapterType
- [uint16](#) hwVersion
- [VixDiskLibSectorType](#) capacity
- [uint32](#) logicalSectorSize
- [uint32](#) physicalSectorSize

3.7.1 Field Documentation

3.7.1.1 VixDiskLibDiskType VixDiskLibCreateParams::diskType

3.7.1.2 VixDiskLibAdapterType VixDiskLibCreateParams::adapterType

3.7.1.3 uint16 VixDiskLibCreateParams::hwVersion

3.7.1.4 VixDiskLibSectorType VixDiskLibCreateParams::capacity

3.7.1.5 uint32 VixDiskLibCreateParams::logicalSectorSize

3.7.1.6 uint32 VixDiskLibCreateParams::physicalSectorSize

3.8 VixDiskLibDatastoreSpec Struct Reference

Data Fields

- char * [datastoreMoRef](#)

3.8.1 Field Documentation

3.8.1.1 char* VixDiskLibDatastoreSpec::datastoreMoRef

3.9 VixDiskLibGeometry Struct Reference

Data Fields

- [uint32 cylinders](#)
- [uint32 heads](#)
- [uint32 sectors](#)

3.9.1 Field Documentation

3.9.1.1 `uint32 VixDiskLibGeometry::cylinders`

3.9.1.2 `uint32 VixDiskLibGeometry::heads`

3.9.1.3 `uint32 VixDiskLibGeometry::sectors`

3.10 VixDiskLibInfo Struct Reference

Data Fields

- [VixDiskLibGeometry biosGeo](#)
- [VixDiskLibGeometry physGeo](#)
- [VixDiskLibSectorType capacity](#)
- [VixDiskLibAdapterType adapterType](#)
- int numLinks
- char * parentFileNameHint
- char * uuid
- uint32 logicalSectorSize
- uint32 physicalSectorSize

3.10.1 Field Documentation

3.10.1.1 VixDiskLibGeometry VixDiskLibInfo::biosGeo

3.10.1.2 VixDiskLibGeometry VixDiskLibInfo::physGeo

3.10.1.3 VixDiskLibSectorType VixDiskLibInfo::capacity

3.10.1.4 VixDiskLibAdapterType VixDiskLibInfo::adapterType

3.10.1.5 int VixDiskLibInfo::numLinks

3.10.1.6 char* VixDiskLibInfo::parentFileNameHint

3.10.1.7 char* VixDiskLibInfo::uuid

3.10.1.8 uint32 VixDiskLibInfo::logicalSectorSize

3.10.1.9 uint32 VixDiskLibInfo::physicalSectorSize

3.11 VixDiskLibNasPlugin Struct Reference

Data Fields

- [VixDiskLibPlugin](#) `diskLibPlugin`
- [VixDiskLibNasPluginStartSession](#) * `StartSession`
- [VixDiskLibNasPluginEndSession](#) * `EndSession`
- [VixDiskLibNasPluginExecutePrimitive](#) * `ExecPrimitive`
- [VixDiskLibNasPluginSupportStatus](#) * `SupportStatus`

3.11.1 Detailed Description

Plugin function table to be exported by a NAS plugin. See the description of the various data types for explanation.

3.11.2 Field Documentation

3.11.2.1 `VixDiskLibPlugin` `VixDiskLibNasPlugin::diskLibPlugin`

3.11.2.2 `VixDiskLibNasPluginStartSession*` `VixDiskLibNasPlugin::StartSession`

3.11.2.3 `VixDiskLibNasPluginEndSession*` `VixDiskLibNasPlugin::EndSession`

3.11.2.4 `VixDiskLibNasPluginExecutePrimitive*` `VixDiskLibNasPlugin::ExecPrimitive`

3.11.2.5 `VixDiskLibNasPluginSupportStatus*` `VixDiskLibNasPlugin::SupportStatus`

3.12 VixDiskLibNasPluginCloneFileParams Struct Reference

Data Fields

- [VixDiskLibNasPluginCommonParams](#) common
- char * srcFileName
- char * dstFileName
- [VixDiskLibNasPluginCloneFileFlags](#) cloneFlags
- [VixDiskLibNasPluginDataStoreParams](#) * srcDataStoreInfo

3.12.1 Field Documentation

3.12.1.1 [VixDiskLibNasPluginCommonParams](#) [VixDiskLibNasPluginCloneFileParams::common](#)

3.12.1.2 char* [VixDiskLibNasPluginCloneFileParams::srcFileName](#)

3.12.1.3 char* [VixDiskLibNasPluginCloneFileParams::dstFileName](#)

3.12.1.4 [VixDiskLibNasPluginCloneFileFlags](#) [VixDiskLibNasPluginCloneFileParams::cloneFlags](#)

3.12.1.5 [VixDiskLibNasPluginDataStoreParams](#)* [VixDiskLibNasPluginCloneFileParams::srcDataStoreInfo](#)

3.13 VixDiskLibNasPluginCommonParams Struct Reference

Data Fields

- [VixDiskLibNasPluginPrimitiveID](#) primitiveID
- [VixDiskLibNasPluginProgressRecord](#) * progressRecord
- [VixDiskLibNasPluginResultCommon](#) * result

3.13.1 Field Documentation

3.13.1.1 VixDiskLibNasPluginPrimitiveID VixDiskLibNasPluginCommonParams::primitiveID

3.13.1.2 VixDiskLibNasPluginProgressRecord* VixDiskLibNasPluginCommonParams::progressRecord

3.13.1.3 VixDiskLibNasPluginResultCommon* VixDiskLibNasPluginCommonParams::result

3.14 VixDiskLibNasPluginDataStoreParams Struct Reference

Data Fields

- char * [fsType](#)
- uint32 [fsVersion](#)
- char * [remoteIP](#)
- char * [remoteMountPoint](#)
- char * [localMountPoint](#)

3.14.1 Field Documentation

3.14.1.1 char* VixDiskLibNasPluginDataStoreParams::fsType

3.14.1.2 uint32 VixDiskLibNasPluginDataStoreParams::fsVersion

3.14.1.3 char* VixDiskLibNasPluginDataStoreParams::remoteIP

3.14.1.4 char* VixDiskLibNasPluginDataStoreParams::remoteMountPoint

3.14.1.5 char* VixDiskLibNasPluginDataStoreParams::localMountPoint

3.15 VixDiskLibNasPluginProgressRecord Struct Reference

Data Fields

- void * [private](#)
- uint64 [updateBytes](#)
- uint64 [progressBytes](#)
- [VixDiskLibNasPluginPeriodicCallback](#) * [callback](#)

3.15.1 Field Documentation

3.15.1.1 void* [VixDiskLibNasPluginProgressRecord::private](#)

3.15.1.2 uint64 [VixDiskLibNasPluginProgressRecord::updateBytes](#)

3.15.1.3 uint64 [VixDiskLibNasPluginProgressRecord::progressBytes](#)

3.15.1.4 [VixDiskLibNasPluginPeriodicCallback](#)* [VixDiskLibNasPluginProgressRecord::callback](#)

3.16 VixDiskLibNasPluginResultCommon Struct Reference

Data Fields

- VixError [status](#)

3.16.1 Field Documentation

3.16.1.1 VixError VixDiskLibNasPluginResultCommon::status

3.17 VixDiskLibNasPluginResvSpaceParams Struct Reference

Data Fields

- [VixDiskLibNasPluginCommonParams](#) `common`
- `char * fileName`

3.17.1 Field Documentation

3.17.1.1 `VixDiskLibNasPluginCommonParams VixDiskLibNasPluginResvSpaceParams::common`

3.17.1.2 `char* VixDiskLibNasPluginResvSpaceParams::fileName`

3.18 VixDiskLibNasPluginSessionParams Struct Reference

Data Fields

- [uint64 timeoutMS](#)

3.18.1 Field Documentation

3.18.1.1 uint64 VixDiskLibNasPluginSessionParams::timeoutMS

3.19 VixDiskLibNasPluginStatXParams Struct Reference

Data Fields

- [VixDiskLibNasPluginCommonParams](#) `common`
- `char * fileName`

3.19.1 Field Documentation

3.19.1.1 `VixDiskLibNasPluginCommonParams VixDiskLibNasPluginStatXParams::common`

3.19.1.2 `char* VixDiskLibNasPluginStatXParams::fileName`

3.20 VixDiskLibNasPluginStatXResult Struct Reference

Data Fields

- [VixDiskLibNasPluginResultCommon](#) common
- [uint64](#) totalBytes
- [uint64](#) allocatedBytes
- [uint64](#) uniqueBytes
- [VixDiskLibNasPluginAllocType](#) allocType

3.20.1 Field Documentation

3.20.1.1 [VixDiskLibNasPluginResultCommon](#) `VixDiskLibNasPluginStatXResult::common`

3.20.1.2 [uint64](#) `VixDiskLibNasPluginStatXResult::totalBytes`

3.20.1.3 [uint64](#) `VixDiskLibNasPluginStatXResult::allocatedBytes`

3.20.1.4 [uint64](#) `VixDiskLibNasPluginStatXResult::uniqueBytes`

3.20.1.5 [VixDiskLibNasPluginAllocType](#) `VixDiskLibNasPluginStatXResult::allocType`

3.21 VixDiskLibPlugin Struct Reference

Data Fields

- int [majorVersion](#)
- int [minorVersion](#)
- [VixDiskLibPluginType](#) type
- const char * [name](#)
- [VixDiskLibPluginInit](#) * [Init](#)
- [VixDiskLibPluginDone](#) * [Done](#)

3.21.1 Detailed Description

Plugin function table to be exported for each plugin the plugin library contains. See the description of the various data types for explanation.

3.21.2 Field Documentation

3.21.2.1 int VixDiskLibPlugin::majorVersion

3.21.2.2 int VixDiskLibPlugin::minorVersion

Major version supported by this plugin.

3.21.2.3 VixDiskLibPluginType VixDiskLibPlugin::type

Minor version supported by this plugin.

3.21.2.4 const char* VixDiskLibPlugin::name

The type of plugin this is

3.21.2.5 VixDiskLibPluginInit* VixDiskLibPlugin::Init

Name associated with this plugin.

3.21.2.6 VixDiskLibPluginDone* VixDiskLibPlugin::Done

Optional

3.22 VixDiskLibSpec Union Reference

Data Fields

- char * [vmxSpec](#)
- [VixDiskLibVStorageObjectSpec](#) vStorageObjSpec
- [VixDiskLibDatastoreSpec](#) dsSpec

3.22.1 Field Documentation

3.22.1.1 char* VixDiskLibSpec::vmxSpec

3.22.1.2 VixDiskLibVStorageObjectSpec VixDiskLibSpec::vStorageObjSpec

3.22.1.3 VixDiskLibDatastoreSpec VixDiskLibSpec::dsSpec

3.23 VixDiskLibVStorageObjectSpec Struct Reference

Data Fields

- char * [id](#)
- char * [datastoreMoRef](#)
- char * [ssId](#)

3.23.1 Field Documentation

3.23.1.1 char* VixDiskLibVStorageObjectSpec::id

3.23.1.2 char* VixDiskLibVStorageObjectSpec::datastoreMoRef

3.23.1.3 char* VixDiskLibVStorageObjectSpec::ssId

3.24 VMPoint Struct Reference

Data Fields

- `int x`
- `int y`

3.24.1 Field Documentation

3.24.1.1 `int VMPoint::x`

3.24.1.2 `int VMPoint::y`

3.25 VMRect Struct Reference

Data Fields

- int [left](#)
- int [top](#)
- int [right](#)
- int [bottom](#)

3.25.1 Field Documentation

3.25.1.1 int VMRect::left

3.25.1.2 int VMRect::top

3.25.1.3 int VMRect::right

3.25.1.4 int VMRect::bottom

Chapter 4

File Documentation

4.1 distribute_vixDiskLib.h File Reference

Data Structures

- struct [VixDiskLibGeometry](#)
- struct [VixDiskLibCreateParams](#)
- struct [VixDiskLibVStorageObjectSpec](#)
- struct [VixDiskLibDatastoreSpec](#)
- union [VixDiskLibSpec](#)
- struct [VixDiskLibConnectParams](#)
- union [VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds](#)
- struct [VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds](#)
- struct [VixDiskLibConnectParams::VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibConnectParams::VixDiskLibCreds](#)
- struct [VixDiskLibInfo](#)
- struct [VixDiskLibBlock](#)
- struct [VixDiskLibBlockList](#)

Defines

- #define [VIXDISKLIB_SECTOR_SIZE](#) 512
- #define [VIXDISKLIB_HWVERSION_WORKSTATION_4](#) 3
- #define [VIXDISKLIB_HWVERSION_WORKSTATION_5](#) 4
- #define [VIXDISKLIB_HWVERSION_WORKSTATION_6](#) 6
- #define [VIXDISKLIB_HWVERSION_ESX30](#) VIXDISKLIB_HWVERSION_WORKSTATION_5
- #define [VIXDISKLIB_HWVERSION_ESX4X](#) 7
- #define [VIXDISKLIB_HWVERSION_ESX50](#) 8
- #define [VIXDISKLIB_HWVERSION_ESX51](#) 9
- #define [VIXDISKLIB_HWVERSION_ESX55](#) 10
- #define [VIXDISKLIB_HWVERSION_ESX60](#) 11
- #define [VIXDISKLIB_HWVERSION_ESX65](#) 13
- #define [VIXDISKLIB_HWVERSION_CURRENT](#) VIXDISKLIB_HWVERSION_ESX65
- #define [VIXDISKLIB_MIN_CHUNK_SIZE](#) (64 * 2)
- #define [VIXDISKLIB_MAX_CHUNK_SIZE](#) (64 * 2 * 1024)
- #define [VIXDISKLIB_MAX_CHUNK_NUMBER](#) (512 * 1024)

- #define `VIXDISKLIB_FLAG_OPEN_UNBUFFERED` (1 << 0)
- #define `VIXDISKLIB_FLAG_OPEN_SINGLE_LINK` (1 << 1)
- #define `VIXDISKLIB_FLAG_OPEN_READ_ONLY` (1 << 2)
- #define `VIXDISKLIB_FLAG_OPEN_COMPRESSION_ZLIB` (1 << 4)
- #define `VIXDISKLIB_FLAG_OPEN_COMPRESSION_FASTLZ` (1 << 5)
- #define `VIXDISKLIB_FLAG_OPEN_COMPRESSION_SKIPZ` (1 << 6)
- #define `VIXDISKLIB_FLAG_OPEN_COMPRESSION_MASK` (0x03f0)

Typedefs

- typedef `uint64 VixDiskLibSectorType`
- typedef struct `VixDiskLibConnectParamsState VixDiskLibConnectParamsState`
- typedef struct `VixDiskLibHandleStruct VixDiskLibHandleStruct`
- typedef `VixDiskLibHandleStruct * VixDiskLibHandle`
- typedef struct `VixDiskLibConnectParam * VixDiskLibConnection`
- typedef void(`VixDiskLibGenericLogFunc`)(const char *fmt, va_list args)
- typedef void(`VixDiskLibGenericLogVFunc`)(int routing, const char *fmt, va_list args)
- typedef `Bool(* VixDiskLibProgressFunc`)(void *progressData, int percentCompleted)
- typedef void(* `VixDiskLibCompletionCB`)(void *cbData, VixError result)

Enumerations

- enum `VixDiskLibDiskType` {
`VIXDISKLIB_DISK_MONOLITHIC_SPARSE` = 1, `VIXDISKLIB_DISK_MONOLITHIC_FLAT`
= 2, `VIXDISKLIB_DISK_SPLIT_SPARSE` = 3, `VIXDISKLIB_DISK_SPLIT_FLAT` = 4,
`VIXDISKLIB_DISK_VMFS_FLAT` = 5, `VIXDISKLIB_DISK_STREAM_OPTIMIZED` = 6,
`VIXDISKLIB_DISK_VMFS_THIN` = 7, `VIXDISKLIB_DISK_VMFS_SPARSE` = 8,
`VIXDISKLIB_DISK_UNKNOWN` = 256 }
- enum `VixDiskLibAdapterType` { `VIXDISKLIB_ADAPTER_IDE` = 1, `VIXDISKLIB_ADAPTER_-`
`SCSI_BUSLOGIC` = 2, `VIXDISKLIB_ADAPTER_SCSI_LSILOGIC` = 3, `VIXDISKLIB_-`
`ADAPTER_UNKNOWN` = 256 }
- enum `VixDiskLibCredType` {
`VIXDISKLIB_CRED_UID` = 1, `VIXDISKLIB_CRED_SESSIONID` = 2, `VIXDISKLIB_CRED_-`
`TICKETID` = 3, `VIXDISKLIB_CRED_SSPI` = 4,
`VIXDISKLIB_CRED_UNKNOWN` = 256 }
- enum `VixDiskLibSpecType` { `VIXDISKLIB_SPEC_VMX` = 0, `VIXDISKLIB_SPEC_-`
`VSTORAGE_OBJECT` = 1, `VIXDISKLIB_SPEC_DATASTORE` = 128, `VIXDISKLIB_SPEC_-`
`UNKNOWN` = 256 }

Functions

- VixError `VixDiskLib_InitEx` (`uint32` majorVersion, `uint32` minorVersion, `VixDiskLibGenericLogFunc` *log, `VixDiskLibGenericLogFunc` *warn, `VixDiskLibGenericLogFunc` *panic, const char *libDir, const char *configFile)
- VixError `VixDiskLib_Init` (`uint32` majorVersion, `uint32` minorVersion, `VixDiskLibGenericLogFunc` *log, `VixDiskLibGenericLogFunc` *warn, `VixDiskLibGenericLogFunc` *panic, const char *libDir)
- void `VixDiskLib_Exit` (void)
- const char * `VixDiskLib_ListTransportModes` (void)

- VixError `VixDiskLib_Cleanup` (const `VixDiskLibConnectParams` *connectParams, `uint32` *numCleanedUp, `uint32` *numRemaining)
- VixError `VixDiskLib_Connect` (const `VixDiskLibConnectParams` *connectParams, `VixDiskLibConnection` *connection)
- VixError `VixDiskLib_PrepareForAccess` (const `VixDiskLibConnectParams` *connectParams, const char *identity)
- VixError `VixDiskLib_ConnectEx` (const `VixDiskLibConnectParams` *connectParams, `Bool` readOnly, const char *snapshotRef, const char *transportModes, `VixDiskLibConnection` *connection)
- VixError `VixDiskLib_Disconnect` (`VixDiskLibConnection` connection)
- VixError `VixDiskLib_EndAccess` (const `VixDiskLibConnectParams` *connectParams, const char *identity)
- VixError `VixDiskLib_Create` (const `VixDiskLibConnection` connection, const char *path, const `VixDiskLibCreateParams` *createParams, `VixDiskLibProgressFunc` progressFunc, void *progressCallbackData)
- VixError `VixDiskLib_CreateChild` (`VixDiskLibHandle` diskHandle, const char *childPath, `VixDiskLibDiskType` diskType, `VixDiskLibProgressFunc` progressFunc, void *progressCallbackData)
- VixError `VixDiskLib_Open` (const `VixDiskLibConnection` connection, const char *path, `uint32` flags, `VixDiskLibHandle` *diskHandle)
- VixError `VixDiskLib_QueryAllocatedBlocks` (`VixDiskLibHandle` diskHandle, `VixDiskLibSectorType` startSector, `VixDiskLibSectorType` numSectors, `VixDiskLibSectorType` chunkSize, `VixDiskLibBlockList` **blockList)
- VixError `VixDiskLib_FreeBlockList` (`VixDiskLibBlockList` *blockList)
- VixError `VixDiskLib_GetInfo` (`VixDiskLibHandle` diskHandle, `VixDiskLibInfo` **info)
- void `VixDiskLib_FreeInfo` (`VixDiskLibInfo` *info)
- const char * `VixDiskLib_GetTransportMode` (`VixDiskLibHandle` diskHandle)
- VixError `VixDiskLib_Close` (`VixDiskLibHandle` diskHandle)
- VixError `VixDiskLib_Read` (`VixDiskLibHandle` diskHandle, `VixDiskLibSectorType` startSector, `VixDiskLibSectorType` numSectors, `uint8` *readBuffer)
- VixError `VixDiskLib_ReadAsync` (`VixDiskLibHandle` diskHandle, `VixDiskLibSectorType` startSector, `VixDiskLibSectorType` numSectors, `uint8` *readBuffer, `VixDiskLibCompletionCB` callback, void *cbData)
- VixError `VixDiskLib_Write` (`VixDiskLibHandle` diskHandle, `VixDiskLibSectorType` startSector, `VixDiskLibSectorType` numSectors, const `uint8` *writeBuffer)
- VixError `VixDiskLib_WriteAsync` (`VixDiskLibHandle` diskHandle, `VixDiskLibSectorType` startSector, `VixDiskLibSectorType` numSectors, const `uint8` *writeBuffer, `VixDiskLibCompletionCB` callback, void *cbData)
- VixError `VixDiskLib_Flush` (`VixDiskLibHandle` diskHandle)
- VixError `VixDiskLib_Wait` (`VixDiskLibHandle` diskHandle)
- VixError `VixDiskLib_ReadMetadata` (`VixDiskLibHandle` diskHandle, const char *key, char *buf, `size_t` bufLen, `size_t` *requiredLen)
- VixError `VixDiskLib_WriteMetadata` (`VixDiskLibHandle` diskHandle, const char *key, const char *val)
- VixError `VixDiskLib_GetMetadataKeys` (`VixDiskLibHandle` diskHandle, char *keys, `size_t` maxLen, `size_t` *requiredLen)
- VixError `VixDiskLib_Unlink` (`VixDiskLibConnection` connection, const char *path)
- VixError `VixDiskLib_Grow` (`VixDiskLibConnection` connection, const char *path, `VixDiskLibSectorType` capacity, `Bool` updateGeometry, `VixDiskLibProgressFunc` progressFunc, void *progressCallbackData)
- VixError `VixDiskLib_Shrink` (`VixDiskLibHandle` diskHandle, `VixDiskLibProgressFunc` progressFunc, void *progressCallbackData)

- VixError [VixDiskLib_Defragment](#) ([VixDiskLibHandle](#) diskHandle, [VixDiskLibProgressFunc](#) progressFunc, void *progressCallbackData)
- VixError [VixDiskLib_Rename](#) (const char *srcFileName, const char *dstFileName)
- VixError [VixDiskLib_Clone](#) (const [VixDiskLibConnection](#) dstConnection, const char *dstPath, const [VixDiskLibConnection](#) srcConnection, const char *srcPath, const [VixDiskLibCreateParams](#) *vixCreateParams, [VixDiskLibProgressFunc](#) progressFunc, void *progressCallbackData, [Bool](#) overwrite)
- char * [VixDiskLib_GetErrorText](#) (VixError err, const char *locale)
- void [VixDiskLib_FreeErrorText](#) (char *errMsg)
- VixError [VixDiskLib_IsAttachPossible](#) ([VixDiskLibHandle](#) parent, [VixDiskLibHandle](#) child)
- VixError [VixDiskLib_Attach](#) ([VixDiskLibHandle](#) parent, [VixDiskLibHandle](#) child)
- VixError [VixDiskLib_SpaceNeededForClone](#) ([VixDiskLibHandle](#) diskHandle, [VixDiskLibDiskType](#) cloneDiskType, [uint64](#) *spaceNeeded)
- VixError [VixDiskLib_CheckRepair](#) (const [VixDiskLibConnection](#) connection, const char *filename, [Bool](#) repair)
- VixError [VixDiskLib_GetConnectParams](#) (const [VixDiskLibConnection](#) connection, [VixDiskLibConnectParams](#) **connectParams)
- void [VixDiskLib_FreeConnectParams](#) ([VixDiskLibConnectParams](#) *connectParams)
- [VixDiskLibConnectParams](#) * [VixDiskLib_AllocateConnectParams](#) ()

4.1.1 Define Documentation

- 4.1.1.1 `#define VIXDISKLIB_SECTOR_SIZE 512`
- 4.1.1.2 `#define VIXDISKLIB_HWVERSION_WORKSTATION_4 3`
- 4.1.1.3 `#define VIXDISKLIB_HWVERSION_WORKSTATION_5 4`
- 4.1.1.4 `#define VIXDISKLIB_HWVERSION_WORKSTATION_6 6`
- 4.1.1.5 `#define VIXDISKLIB_HWVERSION_ESX30 VIXDISKLIB_HWVERSION_`
`WORKSTATION_5`
- 4.1.1.6 `#define VIXDISKLIB_HWVERSION_ESX4X 7`
- 4.1.1.7 `#define VIXDISKLIB_HWVERSION_ESX50 8`
- 4.1.1.8 `#define VIXDISKLIB_HWVERSION_ESX51 9`
- 4.1.1.9 `#define VIXDISKLIB_HWVERSION_ESX55 10`
- 4.1.1.10 `#define VIXDISKLIB_HWVERSION_ESX60 11`
- 4.1.1.11 `#define VIXDISKLIB_HWVERSION_ESX65 13`
- 4.1.1.12 `#define VIXDISKLIB_HWVERSION_CURRENT VIXDISKLIB_HWVERSION_ESX65`
- 4.1.1.13 `#define VIXDISKLIB_MIN_CHUNK_SIZE (64 * 2)`
- 4.1.1.14 `#define VIXDISKLIB_MAX_CHUNK_SIZE (64 * 2 * 1024)`
- 4.1.1.15 `#define VIXDISKLIB_MAX_CHUNK_NUMBER (512 * 1024)`
- 4.1.1.16 `#define VIXDISKLIB_FLAG_OPEN_UNBUFFERED (1 << 0)`
- 4.1.1.17 `#define VIXDISKLIB_FLAG_OPEN_SINGLE_LINK (1 << 1)`
- 4.1.1.18 `#define VIXDISKLIB_FLAG_OPEN_READ_ONLY (1 << 2)`
- 4.1.1.19 `#define VIXDISKLIB_FLAG_OPEN_COMPRESSION_ZLIB (1 << 4)`
- 4.1.1.20 `#define VIXDISKLIB_FLAG_OPEN_COMPRESSION_FASTLZ (1 << 5)`
- 4.1.1.21 `#define VIXDISKLIB_FLAG_OPEN_COMPRESSION_SKIPZ (1 << 6)`
- 4.1.1.22 `#define VIXDISKLIB_FLAG_OPEN_COMPRESSION_MASK (0x03f0)`

4.1.2 Typedef Documentation

- 4.1.2.1 `typedef uint64 VixDiskLibSectorType`
- 4.1.2.2 `typedef struct VixDiskLibConnectParamsState VixDiskLibConnectParamsState`
- 4.1.2.3 `typedef struct VixDiskLibHandleStruct VixDiskLibHandleStruct`
- 4.1.2.4 `typedef VixDiskLibHandleStruct* VixDiskLibHandle`
- 4.1.2.5 `typedef struct VixDiskLibConnectParam* VixDiskLibConnection`
- 4.1.2.6 `typedef void(VixDiskLibGenericLogFunc)(const char *fmt, va_list args)`

VIXDISKLIB_DISK_MONOLITHIC_FLAT
VIXDISKLIB_DISK_SPLIT_SPARSE
VIXDISKLIB_DISK_SPLIT_FLAT
VIXDISKLIB_DISK_VMFS_FLAT
VIXDISKLIB_DISK_STREAM_OPTIMIZED
VIXDISKLIB_DISK_VMFS_THIN
VIXDISKLIB_DISK_VMFS_SPARSE
VIXDISKLIB_DISK_UNKNOWN

4.1.3.2 enum VixDiskLibAdapterType

Enumerator:

VIXDISKLIB_ADAPTER_IDE
VIXDISKLIB_ADAPTER_SCSI_BUSLOGIC
VIXDISKLIB_ADAPTER_SCSI_LSILOGIC
VIXDISKLIB_ADAPTER_UNKNOWN

4.1.3.3 enum VixDiskLibCredType

Enumerator:

VIXDISKLIB_CRED_UID
VIXDISKLIB_CRED_SESSIONID
VIXDISKLIB_CRED_TICKETID
VIXDISKLIB_CRED_SSPI
VIXDISKLIB_CRED_UNKNOWN

4.1.3.4 enum VixDiskLibSpecType

the type of spec

Enumerator:

VIXDISKLIB_SPEC_VMX
VIXDISKLIB_SPEC_VSTORAGE_OBJECT
VIXDISKLIB_SPEC_DATASTORE
VIXDISKLIB_SPEC_UNKNOWN

4.1.4 Function Documentation

4.1.4.1 `VixError VixDiskLib_InitEx (uint32 majorVersion, uint32 minorVersion, VixDiskLibGenericLogFunc * log, VixDiskLibGenericLogFunc * warn, VixDiskLibGenericLogFunc * panic, const char * libDir, const char * configFile)`

Initializes VixDiskLib.

Parameters:

majorVersion [in] Required major version number for client.
minorVersion [in] Required minor version number for client.
log [in] Callback for Log entries.
warn [in] Callback for warnings.
panic [in] Callback for panic.
libDir [in] Directory location where dependent libs are located.
configFile [in] Configuration file path in local encoding. configuration files are of the format name = "value" each name/value pair on a separate line. For a detailed description of allowed values, refer to the VixDiskLib documentation.

Returns:

VIX_OK on success, suitable VIX error code otherwise.

4.1.4.2 VixError VixDiskLib_Init (uint32 *majorVersion*, uint32 *minorVersion*, VixDiskLibGenericLogFunc * *log*, VixDiskLibGenericLogFunc * *warn*, VixDiskLibGenericLogFunc * *panic*, const char * *libDir*)

Initializes VixDiskLib - deprecated, please use VixDiskLib_InitEx.

Parameters:

majorVersion [in] Required major version number for client.
minorVersion [in] Required minor version number for client.
log [in] Callback for Log entries.
warn [in] Callback for warnings.
panic [in] Callback for panic.
libDir [in] Directory location where dependent libs are located.

Returns:

VIX_OK on success, suitable VIX error code otherwise.

4.1.4.3 void VixDiskLib_Exit (void)

Cleans up VixDiskLib.

4.1.4.4 const char* VixDiskLib_ListTransportModes (void)

Get a list of transport modes known to VixDiskLib. This list is also the default used if VixDiskLib_ConnectEx is called with transportModes set to NULL.

The string is a list of transport modes separated by colons. For example: "file:san:hotadd:nbd". See VixDiskLib_ConnectEx for more details.

Returns:

Returns a string that is a list of plugins. The caller must not free the string.

4.1.4.5 VixError VixDiskLib_Cleanup (const VixDiskLibConnectParams * *connectParams*, uint32 * *numCleanedUp*, uint32 * *numRemaining*)

Perform a cleanup after an unclean shutdown of an application using VixDiskLib.

When using VixDiskLib_ConnectEx, some state might have not been cleaned up if the resulting connection was not shut down cleanly. Use VixDiskLib_Cleanup to remove this extra state.

Parameters:

connectParams [in] Hostname and login credentials to connect to a host managing virtual machines that were accessed and need cleanup. While VixDiskLib_Cleanup can be invoked for local connections as well, it is a no-op in that case. Also, the vmxSpec property of connectParams should be set to NULL.

numCleanedUp [out] Number of virtual machines that were successfully cleaned up. – Can be NULL.

numRemaining [out] Number of virtual machines that still require cleaning up. – Can be NULL.

Returns:

VIX_OK if all virtual machines were successfully cleaned up or if no virtual machines required cleanup. VIX error code otherwise and numRemaining can be used to check for the number of virtual machines requiring cleanup.

4.1.4.6 VixError VixDiskLib_Connect (const VixDiskLibConnectParams * *connectParams*, VixDiskLibConnection * *connection*)

Connects to a local / remote server.

Parameters:

connectParams [in] NULL if manipulating local disks. For remote case this includes esx hostName and user credentials.

connection [out] Returned handle to a connection.

Returns:

VIX_OK if success suitable VIX error code otherwise.

4.1.4.7 VixError VixDiskLib_PrepareForAccess (const VixDiskLibConnectParams * *connectParams*, const char * *identity*)

This function is used to notify the host of the virtual machine that the disks of the virtual machine will be opened. The host disables operations on the virtual machine that may be adversely affected if they are performed while the disks are open by a third party application.

Parameters:

connectParams [in] This is always used on remote connections.

identity [in] An arbitrary string containing the identity of the application.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.8 **VixError VixDiskLib_ConnectEx (const VixDiskLibConnectParams * *connectParams*, Bool *readOnly*, const char * *snapshotRef*, const char * *transportModes*, VixDiskLibConnection * *connection*)**

Create a transport context to access disks belonging to a particular snapshot of a particular virtual machine. Using this transport context will enable callers to open virtual disks using the most efficient data access protocol available for managed virtual machines, hence getting better I/O performance.

If this call is used instead of VixDiskLib_Connect, the additional information passed in will be used in order to optimize the I/O access path, to maximize I/O throughput.

Note: For local virtual machines/disks, this call is equivalent to VixDiskLib_Connect.

Parameters:

connectParams [in] NULL if manipulating local disks. For remote case this includes esx hostName and user credentials.

readOnly [in] Should be set to TRUE if no write access is needed for the disks to be accessed through this connection. In some cases, a more efficient I/O path can be used for read-only access.

snapshotRef [in] A managed object reference to the specific snapshot of the virtual machine whose disks will be accessed with this connection. Specifying this property is only meaningful if the vmxSpec property in connectParams is set as well.

transportModes [in] An optional list of transport modes that can be used for this connection, separated by colons. If NULL is specified, VixDiskLib's default setting of "file:san:hotadd:nbd" is used. If a disk is opened through this connection, VixDiskLib will start with the first entry of the list and attempt to use this transport mode to gain access to the virtual disk. If this does not work, the next item in the list will be used until either the disk was successfully opened or the end of the list is reached.

connection [out] Returned handle to a connection.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.9 **VixError VixDiskLib_Disconnect (VixDiskLibConnection *connection*)**

Breaks an existing connection.

Parameters:

connection [in] Valid handle to a (local/remote) connection.

Returns:

VIX_OK if success suitable VIX error code otherwise.

4.1.4.10 **VixError VixDiskLib_EndAccess (const VixDiskLibConnectParams * *connectParams*, const char * *identity*)**

This function is used to notify the host of a virtual machine that the virtual machine disks are closed and that the operations which rely on the virtual machine disks to be closed can now be allowed.

Parameters:

connectParams [in] Always used for a remote connection. Must be the same parameters as used in the corresponding PrepareForAccess call.

identity [in] An arbitrary string containing the identity of the application.

Returns:

VIX_OK of success, suitable VIX error code otherwise.

4.1.4.11 VixError VixDiskLib_Create (const VixDiskLibConnection *connection*, const char * *path*, const VixDiskLibCreateParams * *createParams*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*)

Creates a local disk. Remote disk creation is not supported.

Parameters:

connection [in] A valid connection.

path [in] VMDK file name given as absolute path e.g. "c:\\My Virtual Machines\\MailServer\SystemDisk.vmdk".

createParams [in] Specification for the new disk (type, capacity ...).

progressFunc [in] Callback to report progress.

progressCallbackData [in] Callback data pointer.

Returns:

VIX_OK if success suitable VIX error code otherwise.

4.1.4.12 VixError VixDiskLib_CreateChild (VixDiskLibHandle *diskHandle*, const char * *childPath*, VixDiskLibDiskType *diskType*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*)

Creates a redo log from a parent disk.

Parameters:

diskHandle [in] Handle to an open virtual disk.

childPath [in] Redo log file name given as absolute path e.g. "c:\\My Virtual Machines\\MailServer\SystemDisk_s0001.vmdk".

diskType [in] Either VIXDISKLIB_DISK_MONOLITHIC_SPARSE or VIXDISKLIB_DISK_SPLIT_SPARSE.

progressFunc [in] Callback to report progress.

progressCallbackData [in] Callback data pointer.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.13 VixError VixDiskLib_Open (const VixDiskLibConnection *connection*, const char * *path*, uint32 *flags*, VixDiskLibHandle * *diskHandle*)

Opens a local or remote virtual disk.

Parameters:

- connection* [in] A valid connection.
- path* [in] VMDK file name given as absolute path e.g. "[storage1] MailServer/SystemDisk.vmdk"
- flags* [in, optional] Bitwise or'ed combination of VIXDISKLIB_FLAG_OPEN_UNBUFFERED VIXDISKLIB_FLAG_OPEN_SINGLE_LINK VIXDISKLIB_FLAG_OPEN_READ_ONLY.
- diskHandle* [out] Handle to opened disk, NULL if disk was not opened.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.14 VixError VixDiskLib_QueryAllocatedBlocks (VixDiskLibHandle *diskHandle*, VixDiskLibSectorType *startSector*, VixDiskLibSectorType *numSectors*, VixDiskLibSectorType *chunkSize*, VixDiskLibBlockList ** *blockList*)

Get the blocks allocated.

Parameters:

- diskHandle* [in] Handle to an open virtual disk.
- startSector* [in] Absolute offset.
- numSectors* [in] Number of sectors to query.
- chunkSize* [in] Minimum number of sectors covered by a chunk containing data.
- blockList* [out] Buffer contains a [VixDiskLibBlockList](#).

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.15 VixError VixDiskLib_FreeBlockList (VixDiskLibBlockList * *blockList*)

Frees memory allocated in VixDiskLib_QueryAllocatedBlocks.

Parameters:

- blockList* [in] block list to be freed.

4.1.4.16 VixError VixDiskLib_GetInfo (VixDiskLibHandle *diskHandle*, VixDiskLibInfo ** *info*)

Retrieves information about a disk.

Parameters:

- diskHandle* [in] Handle to an open virtual disk.
- info* [out] Disk information filled up.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.17 void VixDiskLib_FreeInfo (VixDiskLibInfo * *info*)

Frees memory allocated in VixDiskLib_GetInfo.

Parameters:

info [in] Disk information to be freed.

4.1.4.18 const char* VixDiskLib_GetTransportMode (VixDiskLibHandle *diskHandle*)

Returns a pointer to a static string identifying the transport mode that is used to access the virtual disk's data.

If a disk was opened through a connection obtained by VixDiskLib_Connect, the return value will be "file" for a local disk and "nbd" or "nbdssl" for a managed disk.

The pointer to this string is static and must not be deallocated by the caller.

Parameters:

diskHandle [in] Handle to an open virtual disk.

Returns:

Returns a pointer to a static string identifying the transport mode used to access the disk's data.

4.1.4.19 VixError VixDiskLib_Close (VixDiskLibHandle *diskHandle*)

Closes the disk.

Parameters:

diskHandle [in] Handle to an open virtual disk.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.20 VixError VixDiskLib_Read (VixDiskLibHandle *diskHandle*, VixDiskLibSectorType *startSector*, VixDiskLibSectorType *numSectors*, uint8 * *readBuffer*)

Reads a sector range.

Parameters:

diskHandle [in] Handle to an open virtual disk.

startSector [in] Absolute offset.

numSectors [in] Number of sectors to read.

readBuffer [out] Buffer to read into.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.21 VixError VixDiskLib_ReadAsync (VixDiskLibHandle *diskHandle*, VixDiskLibSectorType *startSector*, VixDiskLibSectorType *numSectors*, uint8 * *readBuffer*, VixDiskLibCompletionCB *callback*, void * *cbData*)

Reads a sector range asynchronously.

Parameters:

diskHandle [in] Handle to an open virtual disk.
startSector [in] Absolute offset.
numSectors [in] Number of sectors to read.
readBuffer [out] Buffer to read into.
callback [in] Callback when data has been read
cbData [in] User context data supplied during the callback

Returns:

VIX_ASYNC if success, suitable VIX error code otherwise.

4.1.4.22 VixError VixDiskLib_Write (VixDiskLibHandle *diskHandle*, VixDiskLibSectorType *startSector*, VixDiskLibSectorType *numSectors*, const uint8 * *writeBuffer*)

Writes a sector range.

Parameters:

diskHandle [in] Handle to an open virtual disk.
startSector [in] Absolute offset.
numSectors [in] Number of sectors to write.
writeBuffer [in] Buffer to write.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.23 VixError VixDiskLib_WriteAsync (VixDiskLibHandle *diskHandle*, VixDiskLibSectorType *startSector*, VixDiskLibSectorType *numSectors*, const uint8 * *writeBuffer*, VixDiskLibCompletionCB *callback*, void * *cbData*)

Writes a sector range asynchronously.

Parameters:

diskHandle [in] Handle to an open virtual disk.
startSector [in] Absolute offset.
numSectors [in] Number of sectors to write.
writeBuffer [in] Buffer to write.
callback [in] Callback when data has been written/completed
cbData [in] User context data supplied during the callback

Returns:

VIX_ASYNC if success, suitable VIX error code otherwise.

4.1.4.24 VixError VixDiskLib_Flush (VixDiskLibHandle *diskHandle*)

Flush the async write data to disk

Parameters:

diskHandle [in] Handle to an open virtual disk.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.25 VixError VixDiskLib_Wait (VixDiskLibHandle *diskHandle*)

Waits for all async operations to complete

Parameters:

diskHandle [in] Handle to an open virtual disk.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.26 VixError VixDiskLib_ReadMetadata (VixDiskLibHandle *diskHandle*, const char * *key*, char * *buf*, size_t *bufLen*, size_t * *requiredLen*)

Retrieves the value of a metadata entry corresponding to the supplied key.

Parameters:

diskHandle [in] Handle to an open virtual disk.

key [in] Key name.

buf [out, optional] Placeholder for key's value in the metadata store, can be NULL.

bufLen [in] Size of the buffer.

requiredLen [out, optional] Size of buffer required for the value (including end of string character)

Returns:

VIX_OK if success, VIX_E_DISK_BUFFER_TOO_SMALL if too small a buffer and other errors as applicable.

4.1.4.27 VixError VixDiskLib_WriteMetadata (VixDiskLibHandle *diskHandle*, const char * *key*, const char * *val*)

Creates or modifies a metadata table entry.

Parameters:

diskHandle [in] Handle to an open virtual disk.

key [in] Key name.

val [in] Key's value.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.28 VixError VixDiskLib_GetMetadataKeys (VixDiskLibHandle *diskHandle*, char * *keys*, size_t *maxLen*, size_t * *requiredLen*)

Retrieves the list of keys in the metadata table. Key names are returned as list of null-terminated strings, followed by an additional NULL character.

Parameters:

diskHandle [in] Handle to an open virtual disk.

keys [out, optional] Keynames buffer, can be NULL.

maxLen [in] Size of the keynames buffer.

requiredLen [out, optional] Space required for the keys including the double end-of-string characters.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.29 VixError VixDiskLib_Unlink (VixDiskLibConnection *connection*, const char * *path*)

Deletes all extents of the specified disk link. If the path refers to a parent disk, the child (redo log) will be orphaned. Unlinking the child does not affect the parent.

Parameters:

connection [in] A valid connection.

path [in] Path to the disk to be deleted.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.30 VixError VixDiskLib_Grow (VixDiskLibConnection *connection*, const char * *path*, VixDiskLibSectorType *capacity*, Bool *updateGeometry*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*)

Grows an existing disk, only local disks are grown.

Precondition:

The specified disk is not open.

Parameters:

connection [in] A valid connection.

path [in] Path to the disk to be grown.

capacity [in] Target size for the disk.

updateGeometry [in] Should vixDiskLib update the geometry?

progressFunc [in] Callback to report progress (called on the same thread).

progressCallbackData [in] Opaque pointer passed along with the percent complete.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.31 VixError VixDiskLib_Shrink (VixDiskLibHandle *diskHandle*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*)

Shrinks an existing disk, only local disks are shrunk.

Parameters:

diskHandle [in] Handle to an open virtual disk.

progressFunc [in] Callback to report progress (called on the same thread).

progressCallbackData [in] Opaque pointer passed along with the percent complete.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.32 VixError VixDiskLib_Defragment (VixDiskLibHandle *diskHandle*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*)

Defragments an existing disk.

Parameters:

diskHandle [in] Handle to an open virtual disk.

progressFunc [in] Callback to report progress (called on the same thread).

progressCallbackData [in] Opaque pointer passed along with the percent complete.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.33 VixError VixDiskLib_Rename (const char * *srcFileName*, const char * *dstFileName*)

Renames a virtual disk.

Parameters:

srcFileName [in] Virtual disk file to rename.

dstFileName [in] New name for the virtual disk.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.34 VixError VixDiskLib_Clone (const VixDiskLibConnection *dstConnection*, const char * *dstPath*, const VixDiskLibConnection *srcConnection*, const char * *srcPath*, const VixDiskLibCreateParams * *vixCreateParams*, VixDiskLibProgressFunc *progressFunc*, void * *progressCallbackData*, Bool *overWrite*)

Copies a disk with proper conversion.

Parameters:

dstConnection [in] A valid connection to access the destination disk.

dstPath [in] Absolute path for the (new) destination disk.

srcConnection [in] A valid connection to access the source disk.

srcPath [in] Absolute path for the source disk.

vixCreateParams [in] creationParameters (disktype, hardware type...). If the destination is remote, createParams is currently ignored and disk with default size and adapter type is created.

progressFunc [in] Callback to report progress (called on the same thread).

progressCallbackData [in] Opaque pointer passed along with the percent complete.

overWrite [in] TRUE if Clone should overwrite an existing file.

Returns:

VIX_OK if success, suitable VIX error code otherwise (network errors like file already exists handshake failure, ... are all combined into a generic connect message).

4.1.4.35 char* VixDiskLib_GetErrorText (VixError *err*, const char * *locale*)

Returns the textual description of an error.

Parameters:

err [in] A VIX error code.

locale [in] Language locale - not currently supported and must be NULL.

Returns:

The error message string. This should only be deallocated by VixDiskLib_FreeErrorText. Returns NULL if there is an error in retrieving text.

4.1.4.36 void VixDiskLib_FreeErrorText (char * *errMsg*)

Free the error message returned by VixDiskLib_GetErrorText.

Parameters:

errMsg [in] Message string returned by VixDiskLib_GetErrorText. It is OK to call this function with NULL.

Returns:

None.

4.1.4.37 VixError VixDiskLib_IsAttachPossible (VixDiskLibHandle *parent*, VixDiskLibHandle *child*)

Checks if the child disk chain can be attached to the parent disk chain.

Parameters:

parent [in] Handle to the disk to be attached.

child [in] Handle to the disk to attach.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.38 VixError VixDiskLib_Attach (VixDiskLibHandle *parent*, VixDiskLibHandle *child*)

Attaches the child disk chain to the parent disk chain. Parent handle is invalid after attaching and child represents the combined disk chain.

Parameters:

parent [in] Handle to the disk to be attached.

child [in] Handle to the disk to attach.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.39 VixError VixDiskLib_SpaceNeededForClone (VixDiskLibHandle *diskHandle*, VixDiskLibDiskType *cloneDiskType*, uint64 * *spaceNeeded*)

Compute the space (in bytes) required to copy a disk chain.

Parameters:

diskHandle [in] Handle to the disk to be copied.

cloneDiskType [in] Type of the (to be) newly created disk. If cloneDiskType is VIXDISKLIB_DISK_UNKNOWN, the source disk type is assumed.

spaceNeeded [out] Place holder for space needed in bytes.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.40 VixError VixDiskLib_CheckRepair (const VixDiskLibConnection *connection*, const char * *filename*, Bool *repair*)

Check a sparse disk for internal consistency.

Parameters:

connection [in] A VixDiskLib connection.

filename [in] Path to disk to be checked.

repair [in] TRUE if repair should be attempted, false otherwise.

Returns:

VIX_OK if success, suitable VIX error code otherwise. Note this refers to the success of the call, not the consistency of the disk being checked.

4.1.4.41 VixError VixDiskLib_GetConnectParams (const VixDiskLibConnection *connection*, VixDiskLibConnectParams ** *connectParams*)

Return the details for the connection.

Parameters:

connection [in] A VixDiskLib connection.

connectParams [out] Details of the connection.

Returns:

VIX_OK if success, suitable VIX error code otherwise.

4.1.4.42 void VixDiskLib_FreeConnectParams (VixDiskLibConnectParams * *connectParams*)

Free the connection details structure allocated during VixDiskLib_GetConnectParams or VixDiskLib_AllocateConnectParams.

Parameters:

connectParams [out] Connection details to be free'ed.

Returns:

None.

4.1.4.43 VixDiskLibConnectParams* VixDiskLib_AllocateConnectParams ()

Allocate the connection details structure

Returns:

A pointer to the instance of connection details; NULL if error.

4.2 distribute_vixDiskLibNasPlugin.h File Reference

Data Structures

- struct [VixDiskLibNasPluginDataStoreParams](#)
- struct [VixDiskLibNasPluginSessionParams](#)
- struct [VixDiskLibNasPluginProgressRecord](#)
- struct [VixDiskLibNasPluginResultCommon](#)
- struct [VixDiskLibNasPluginStatXResult](#)
- struct [VixDiskLibNasPluginCommonParams](#)
- struct [VixDiskLibNasPluginCloneFileParams](#)
- struct [VixDiskLibNasPluginResvSpaceParams](#)
- struct [VixDiskLibNasPluginStatXParams](#)
- struct [VixDiskLibNasPlugin](#)

Defines

- #define [VIXDISKLIB_NASPLUGIN_MAJOR_VERSION](#) 1
- #define [VIXDISKLIB_NASPLUGIN_MINOR_VERSION](#) 2
- #define [VIXDISKLIB_NASPLUGIN_FSTYPE_NFS](#) "NFS"
- #define [VIXDISKLIB_NASPLUGIN_FSTYPE_NFS41](#) "NFS41"
- #define [VIXDISKLIB_NASPLUGIN_FSTYPE_VMFS](#) "VMFS"
- #define [VIXDISKLIB_NASPLUGIN_FSTYPE_VMFSL](#) "VMFS-L"
- #define [VIXDISKLIB_NASPLUGIN_INVALID_SESSION_ID](#) (0LL)

Typedefs

- typedef uint64 [VixDiskLibNasPluginSessionID](#)
- typedef Bool([VixDiskLibNasPluginPeriodicCallback](#))(struct [VixDiskLibNasPluginProgressRecord](#) *pRec)
- typedef VixError([VixDiskLibNasPluginStartSession](#))(const [VixDiskLibNasPluginDataStoreParams](#) *nasConfigData, const [VixDiskLibNasPluginSessionParams](#) *sessionParams, [VixDiskLibNasPluginSessionID](#) *sessionID)
- typedef VixError([VixDiskLibNasPluginEndSession](#))(VixDiskLibNasPluginSessionID sessionID)
- typedef VixError([VixDiskLibNasPluginSupportStatus](#))(VixDiskLibNasPluginSessionID sessionID, const [VixDiskLibNasPluginPrimitiveID](#) primitiveID)
- typedef void([VixDiskLibNasPluginExecutePrimitive](#))(VixDiskLibNasPluginSessionID sessionID, const [VixDiskLibNasPluginCommonParams](#) *execParams)

Enumerations

- enum [VixDiskLibNasPluginPrimitiveID](#) { [VIXDISKLIB_NASPLUGIN_PRIM_INVALID](#) = (('N' << 24) | ('A' << 16) | ('S' << 8) | '0'), [VIXDISKLIB_NASPLUGIN_PRIM_CLONEFILE](#), [VIXDISKLIB_NASPLUGIN_PRIM_RESVSPACE](#), [VIXDISKLIB_NASPLUGIN_PRIM_STATX](#) }
- enum [VixDiskLibNasPluginAllocType](#) { [VIXDISKLIB_NASPLUGIN_FILE_ALLOC_EZT](#), [VIXDISKLIB_NASPLUGIN_FILE_ALLOC_LZT](#), [VIXDISKLIB_NASPLUGIN_FILE_ALLOC_THIN](#), [VIXDISKLIB_NASPLUGIN_FILE_ALLOC_UNKNOWN](#) = -1 }

- enum `VixDiskLibNasPluginCloneFileFlags` {
`VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_GUARDED = (1 << 0)`, `VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_LAZY = (1 << 1)`, `VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_SRCDATASTORE_VALID = (1 << 2)`, `VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_DRYRUN = (1 << 3)`,
`VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_SKIPZEROES = (1 << 4)` }

4.2.1 Define Documentation

4.2.1.1 `#define VIXDISKLIB_NASPLUGIN_MAJOR_VERSION 1`

4.2.1.2 `#define VIXDISKLIB_NASPLUGIN_MINOR_VERSION 2`

4.2.1.3 `#define VIXDISKLIB_NASPLUGIN_FSTYPE_NFS "NFS"`

4.2.1.4 `#define VIXDISKLIB_NASPLUGIN_FSTYPE_NFS41 "NFS41"`

4.2.1.5 `#define VIXDISKLIB_NASPLUGIN_FSTYPE_VMFS "VMFS"`

4.2.1.6 `#define VIXDISKLIB_NASPLUGIN_FSTYPE_VMFSL "VMFS-L"`

4.2.1.7 `#define VIXDISKLIB_NASPLUGIN_INVALID_SESSION_ID (0LL)`

4.2.2 Typedef Documentation

4.2.2.1 `typedef uint64 VixDiskLibNasPluginSessionID`

4.2.2.2 `typedef Bool(VixDiskLibNasPluginPeriodicCallback)(struct VixDiskLibNasPluginProgressRecord *pRec)`

4.2.2.3 `typedef VixError(VixDiskLibNasPluginStartSession)(const VixDiskLibNasPluginDataStoreParams *nasConfigData, const VixDiskLibNasPluginSessionParams *sessionParams, VixDiskLibNasPluginSessionID *sessionID)`

Prototype for [VixDiskLibNasPlugin](#) session initialization.

The NAS plugin needs configuration information to establish connectivity to the NAS device. VI already stores this information as part of NFS datastore configuration for VMFS. The NAS plugin is provided NAS configuration information as part of an init session request. The configuration data contains the NAS device IP address, NAS device mount point, corresponding datastore name on the ESX host, and the ESX filesystem type and version. This should enable a NAS plugin writer to connect to, say, a RPC program running on the NAS device and exchange other vendor-specific setup information and configuration data. The plugin should return a session identifier for the virtual disk library to use as a context for a single or a set of VM operations. The client can also negotiate session properties like timeouts, priority etc. during session initiation. The session initialization may fail if the plugin does not support the given NAS volume.

Note that this entry point may be called for a datastore which the plugin does not support. For example, it may be called for a datastore on a NAS server from a different vendor, or datastore on a NAS server whose firmware does not support NAS offloads. It is the plugin's responsibility to verify it can communicate with the NAS server and that the NAS server supports the offload primitives and that the remote mount point exists on the NAS server.

Parameters:

- nasConfigData* IN The configuration data contains the NAS device IP address, NAS device mount point, corresponding datastore name on the ESX host, and the ESX filesystem type and version.
- flags* IN sessionParams Session properties like timeouts, priority
- sessionID* OUT Structure returned in the case of a successful session; it is an identifier for the virtual disk library to use as a context for a single or a set of VM operations.

Returns:

VIX_OK if the session was successfully established, a VIX_* error otherwise.

4.2.2.4 typedef VixError(VixDiskLibNasPluginEndSession)(VixDiskLibNasPluginSessionID sessionID)

Prototype for [VixDiskLibNasPlugin](#) session teardown.

The caller may issue multiple sets of VixDiskLibNasPluginExecutePrimitive calls in a given session. A session may end at the request of the caller or after the plugin hits a timeout.

Parameters:

- VixDiskLibNasPluginSessionID* sessionID IN Structure that was returned by the corresponding VixDiskLibNasPluginStartSession call.

Returns:

VIX_OK if the session was successfully terminated, a VIX_* error otherwise.

4.2.2.5 typedef VixError(VixDiskLibNasPluginSupportStatus)(VixDiskLibNasPluginSessionID sessionID, const VixDiskLibNasPluginPrimitiveID primitiveID)

Prototype for [VixDiskLibNasPlugin](#) entrypoint to check if a primitive is supported.

Once a session is established, the caller can ask for the support status of a vStorage primitive in the context of an established session ID. For example, the caller may ask for support status of the VIXDISKLIB_-NASPLUGIN_PRIM_STATX primitive. The plugin shall determine if the offload primitive is supported on the datastore. This may require communication with the NAS device, and the method is vendor specific.

Parameters:

- sessionID* IN Handle identifying a live session
- primitiveID* IN Primitive ID to get the support status for.

Returns:

VIX_OK if the session support was successfully determined, a VIX_* error otherwise.

4.2.2.6 typedef void(VixDiskLibNasPluginExecutePrimitive)(VixDiskLibNasPluginSessionID sessionID, const VixDiskLibNasPluginCommonParams *execParams)

Prototype for [VixDiskLibNasPlugin](#) primitive execution.

Once a session is established, the caller can ask for a vStorage primitive to be done on a set of files in the context of an established session ID. For example, the caller may ask for a file to be quick-cloned (snapshotted) to another file on the same datastore. The plugin shall issue the request to the NAS device in a vendor specific format.

One example of a primitive that would be very useful on NFS is the ability to reserve physical space to back the logical size of a file. The preferred disk format to run enterprise workloads in ESX virtual machines is zeroedthick; VMFS-3 reserves space for zeroedthick virtual disk files at file creation time, thus removing the possibility of the virtual machine crashing because of lack of physical space at the time the application writes to the virtual disk. A newly created virtual disk file on a NFS server could be set to an arbitrary length (many tens or hundreds of GB in this case) using POSIX lseek and write, but the NFS server will not allocate backing storage to the file. This violates the space guarantee requirement of zeroedthick disk format. On NAS devices that understand the vStorage API for file devices, one can issue an VixDiskLibNasPluginExecutePrimitive with primitiveID set to VIXDISKLIB_NASPLUGIN_PRIM_RESVSPACE to instruct the NAS device to use vendor-specific mechanisms to reserve space for a newly created or existing virtual disk of non-zero logical size.

This endpoint can block until the client request is successfully completed on the NAS device. However, the plugin must periodically invoke the provided update callback. Specifically, the plugin must invoke `execParams->progressRecord->callback` for every `execParams->progressRecord->updateBytes` bytes that it transfers, each time setting the `execParams->progressRecord->progressBytes` appropriately. If the callback returns FALSE, it is an indication that the user has requested to abort the operation; as such, the plugin must abort the operation (including the cleanup of any internal state), set the `execParams->result` data appropriately, and return. If communication with the NAS device is broken or timed out or no progress is reported by the NAS device within the session timeout period, the NAS plugin must abort the operation and return.

Parameters:

sessionID IN Handle identifying a live session
execParams IN Parameters to the operation.

4.2.3 Enumeration Type Documentation

4.2.3.1 enum VixDiskLibNasPluginPrimitiveID

Enumerator:

VIXDISKLIB_NASPLUGIN_PRIM_INVALID
VIXDISKLIB_NASPLUGIN_PRIM_CLONEFILE
VIXDISKLIB_NASPLUGIN_PRIM_RESVSPACE
VIXDISKLIB_NASPLUGIN_PRIM_STATX

4.2.3.2 enum VixDiskLibNasPluginAllocType

Enumerator:

VIXDISKLIB_NASPLUGIN_FILE_ALLOC_EZT
VIXDISKLIB_NASPLUGIN_FILE_ALLOC_LZT
VIXDISKLIB_NASPLUGIN_FILE_ALLOC_THIN
VIXDISKLIB_NASPLUGIN_FILE_ALLOC_UNKNOWN

4.2.3.3 enum VixDiskLibNasPluginCloneFileFlags

Enumerator:

VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_GUARDED
VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_LAZY
VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_SRCDATASTORE_VALID
VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_DRYRUN
VIXDISKLIB_NASPLUGIN_CLONEFILE_FLAG_SKIPZEROES

4.3 distribute_vixDiskLibPlugin.h File Reference

Data Structures

- struct [VixDiskLibPlugin](#)

Defines

- #define [VIXDISKLIB_PLUGIN_MAJOR_VERSION](#) 1
- #define [VIXDISKLIB_PLUGIN_MINOR_VERSION](#) 0

Typedefs

- typedef [VixError](#)([VixDiskLibPluginInit](#))([VixDiskLibGenericLogFunc](#) *log, [VixDiskLibGenericLogFunc](#) *warn, [VixDiskLibGenericLogFunc](#) *panic)
- typedef void([VixDiskLibPluginDone](#))(void)

Enumerations

- enum [VixDiskLibPluginType](#) { [VIXDISKLIB_PLUGIN_TYPE_TRANSPORT](#), [VIXDISKLIB_PLUGIN_TYPE_NAS](#), [VIXDISKLIB_PLUGIN_TYPE_TRANSPORT_NO_UNLOAD](#) }

Variables

- [VixDiskLibPlugin](#) ** [VixDiskLibPlugin_EntryPoint](#)

4.3.1 Define Documentation

4.3.1.1 #define [VIXDISKLIB_PLUGIN_MAJOR_VERSION](#) 1

4.3.1.2 #define [VIXDISKLIB_PLUGIN_MINOR_VERSION](#) 0

4.3.2 Typedef Documentation

4.3.2.1 typedef [VixError](#)([VixDiskLibPluginInit](#))([VixDiskLibGenericLogFunc](#) *log, [VixDiskLibGenericLogFunc](#) *warn, [VixDiskLibGenericLogFunc](#) *panic)

Prototype for initializing a plugin in the library. This function will be called when the plugin is loaded to initialize the plugin. If anything but VIX_OK is returned, the plugin will not be loaded.

Parameters:

log IN Log function.

log IN Warning function.

log IN Panic function.

4.3.2.2 typedef void(VixDiskLibPluginDone)(void)

Function to be called when a plugin is unloaded.

4.3.3 Enumeration Type Documentation

4.3.3.1 enum VixDiskLibPluginType

Enumerator:

VIXDISKLIB_PLUGIN_TYPE_TRANSPORT

VIXDISKLIB_PLUGIN_TYPE_NAS

VIXDISKLIB_PLUGIN_TYPE_TRANSPORT_NO_UNLOAD

4.3.4 Variable Documentation

4.3.4.1 VixDiskLibPlugin** VixDiskLibPlugin_EntryPoint

Main entry point into plugin that must be exported. This is an array of pointers to VixDiskLibPlugin structures corresponding to the plugins in this library. The last element in the array must be a NULL pointer.

4.4 public_vm_basic_types.h File Reference

Data Structures

- struct [VMPoint](#)
- struct [VMRect](#)

Defines

- #define [INCLUDE_ALLOW_USERLEVEL](#)
- #define [INCLUDE_ALLOW_MODULE](#)
- #define [INCLUDE_ALLOW_VMMON](#)
- #define [INCLUDE_ALLOW_VMKERNEL](#)
- #define [INCLUDE_ALLOW_VMKDRIVERS](#)
- #define [INCLUDE_ALLOW_VMK_MODULE](#)
- #define [INCLUDE_ALLOW_DISTRIBUTE](#)
- #define [INCLUDE_ALLOW_VMCORE](#)
- #define [vm_x86_64](#) 0
- #define [vm_arm_64](#) 0
- #define [vm_64bit](#) (sizeof (void *) == 8)
- #define [_XTYPEDEF_BOOL](#)
- #define [FALSE](#) 0
- #define [TRUE](#) 1
- #define [IS_BOOL](#)(x) (((x) & ~1) == 0)
- #define [CONST3264](#)(a) (a)
- #define [CONST3264U](#)(a) (a)
- #define [MIN_INT8](#) ((int8)0x80)
- #define [MAX_INT8](#) ((int8)0x7f)
- #define [MIN_UINT8](#) ((uint8)0)
- #define [MAX_UINT8](#) ((uint8)0xff)
- #define [MIN_INT16](#) ((int16)0x8000)
- #define [MAX_INT16](#) ((int16)0x7fff)
- #define [MIN_UINT16](#) ((uint16)0)
- #define [MAX_UINT16](#) ((uint16)0xffff)
- #define [MIN_INT32](#) ((int32)0x80000000)
- #define [MAX_INT32](#) ((int32)0x7fffffff)
- #define [MIN_UINT32](#) ((uint32)0)
- #define [MAX_UINT32](#) ((uint32)0xffffffff)
- #define [MIN_INT64](#) (CONST64(0x8000000000000000))
- #define [MAX_INT64](#) (CONST64(0x7fffffffffffffff))
- #define [MIN_UINT64](#) (CONST64U(0))
- #define [MAX_UINT64](#) (CONST64U(0xfffffffffffffff))
- #define [AsPercent](#)(v) ((Percent)(v))
- #define [UINT64_2_BPN](#)(u) ((BPN)(u))
- #define [BPN_2_UINT64](#)(b) ((uint64)(b))
- #define [INVALID_WORLD_ID](#) ((World_ID)0)
- #define [INVALID_CARTEL_ID](#) INVALID_WORLD_ID
- #define [INVALID_SESSION_ID](#) INVALID_CARTEL_ID
- #define [INVALID_CARTELGROUP_ID](#) INVALID_CARTEL_ID
- #define [INVALID_WORLDLET_ID](#) ((Worldlet_ID)-1)

- #define `LA_2_LPN(_la)` $((_la) \gg \text{PAGE_SHIFT})$
- #define `LPN_2_LA(_lpn)` $((_lpn) \ll \text{PAGE_SHIFT})$
- #define `LAST_LPN` $((((\text{LA}) 1) \ll (8 * \text{sizeof}(\text{LA}) - \text{PAGE_SHIFT})) - 1)$
- #define `LAST_LPN32` $((((\text{LA32}) 1) \ll (8 * \text{sizeof}(\text{LA32}) - \text{PAGE_SHIFT})) - 1)$
- #define `LAST_LPN64` $((((\text{LA64}) 1) \ll (8 * \text{sizeof}(\text{LA64}) - \text{PAGE_SHIFT})) - 1)$
- #define `LPN_MASK` `LAST_LPN`
- #define `LPN_MASK32` `LAST_LPN32`
- #define `LPN_MASK64` `LAST_LPN64`
- #define `MAX_PPN_BITS` `33`
- #define `MAX_PPN` $((\text{PPN}) 1 \ll \text{MAX_PPN_BITS}) - 1$
- #define `INVALID_PPN` $((\text{PPN}) 0 \times 000\text{fffffffffffull})$
- #define `INVALID_PPN32` $((\text{PPN32}) 0 \times \text{fffffff})$
- #define `APIC_INVALID_PPN` $((\text{PPN}) 0 \times 000\text{ffffffffffeull})$
- #define `INVALID_BPN` $((\text{BPN}) 0 \times 0000\text{fffffffffffull})$
- #define `MPN38_MASK` $((1 \text{ull} \ll 38) - 1)$
- #define `RESERVED_MPN` $((\text{MPN}) 0)$
- #define `INVALID_MPN` $((\text{MPN}) \text{MPN38_MASK})$
- #define `MEMREF_MPN` $((\text{MPN}) \text{MPN38_MASK} - 1)$
- #define `RELEASED_MPN` $((\text{MPN}) \text{MPN38_MASK} - 2)$
- #define `MAX_MPN` $((\text{MPN}) \text{MPN38_MASK} - 3)$
- #define `INVALID_IOPN` $((\text{IOPN}) - 1)$
- #define `MAX_IOPN` $(\text{INVALID_IOPN} - 1)$
- #define `INVALID_LPN` $((\text{LPN}) - 1)$
- #define `INVALID_VPN` $((\text{VPN}) - 1)$
- #define `INVALID_LPN64` $((\text{LPN64}) - 1)$
- #define `INVALID_PAGENUM` $((\text{PageNum}) 0 \times 000000\text{fffffffffffull})$
- #define `INVALID_PAGENUM32` $((\text{uint32}) - 1)$
- #define `FMTLA` `""`
- #define `FMTVA` `""`
- #define `FMTVPN` `""`
- #define `EXTERN` `extern`
- #define `CONST` `const`
- #define `INLINE` `inline`
- #define `VMX86_EXTERN_DATA` `extern`
- #define `INLINE_ALWAYS` `INLINE`
- #define `INLINE_SINGLE_CALLER` `INLINE_ALWAYS`
- #define `SIDE_EFFECT_FREE`
- #define `CONST_FUNCTION`
- #define `NORETURN`
- #define `HOT`
- #define `COLD`
- #define `LIKELY(_exp)` $(_exp)$
- #define `UNLIKELY(_exp)` $(_exp)$
- #define `PRINTF_DECL(fmtPos, varPos)`
- #define `SCANF_DECL(fmtPos, varPos)`
- #define `UNUSED_PARAM(_parm)` $_parm$
- #define `UNUSED_TYPE(_parm)` `UNUSED_PARAM(_parm)`
- #define `UNUSED_VARIABLE(_var)` $(\text{void})_var$
- #define `MUST_CHECK_RETURN`
- #define `ALIGNED(n)`

- #define [INFINITE_LOOP\(\)](#) do { } while (1)
- #define [FMTPID](#) "d"
- #define [FMTUID](#) "u"
- #define [FMTMODE](#) "o"

Typedefs

- typedef uint64_t [uint64](#)
- typedef int64_t [int64](#)
- typedef uint32_t [uint32](#)
- typedef int32_t [int32](#)
- typedef uint16_t [uint16](#)
- typedef int16_t [int16](#)
- typedef uint8_t [uint8](#)
- typedef int8_t [int8](#)
- typedef char [Bool](#)
- typedef [int64](#) [VmTimeType](#)
- typedef [int64](#) [VmTimeRealClock](#)
- typedef [int64](#) [VmTimeVirtualClock](#)
- typedef [uint8](#) * [TCA](#)
- typedef [uint8](#) [Percent](#)
- typedef uintptr_t [VA](#)
- typedef uintptr_t [VPN](#)
- typedef [uint64](#) [PA](#)
- typedef [uint64](#) [PPN](#)
- typedef [uint64](#) [TPA](#)
- typedef [uint64](#) [TPPN](#)
- typedef [uint64](#) [PhysMemOff](#)
- typedef [uint64](#) [PhysMemSize](#)
- typedef [uint64](#) [BA](#)
- typedef [uint64](#) [BPN](#)
- typedef [uint64](#) [PageCnt](#)
- typedef [uint64](#) [PageNum](#)
- typedef unsigned [MemHandle](#)
- typedef unsigned [IoHandle](#)
- typedef [int32](#) [World_ID](#)
- typedef [World_ID](#) [User_CartelID](#)
- typedef [User_CartelID](#) [User_SessionID](#)
- typedef [User_CartelID](#) [User_CartelGroupID](#)
- typedef [uint32](#) [Worldlet_ID](#)
- typedef [int8](#) [Reg8](#)
- typedef [int16](#) [Reg16](#)
- typedef [int32](#) [Reg32](#)
- typedef [int64](#) [Reg64](#)
- typedef [uint8](#) [UReg8](#)
- typedef [uint16](#) [UReg16](#)
- typedef [uint32](#) [UReg32](#)
- typedef [uint64](#) [UReg64](#)
- typedef [uint64](#) [MA](#)
- typedef [uint32](#) [MPN32](#)

- typedef uint64 SectorType
- typedef uintptr_t LA
- typedef uintptr_t LPN
- typedef uint32 VA32
- typedef uint32 VPN32
- typedef uint32 LA32
- typedef uint32 LPN32
- typedef uint32 PA32
- typedef uint32 PPN32
- typedef uint64 VA64
- typedef uint64 VPN64
- typedef uint64 LA64
- typedef uint64 LPN64
- typedef uint64 PA64
- typedef uint64 PPN64
- typedef uint64 TPPN64
- typedef uint64 MA64
- typedef uint64 MPN
- typedef uint64 IOA
- typedef uint64 IOPN
- typedef VA32 UserVA32
- typedef VA64 UserVA64
- typedef UserVA64 UserVAConst
- typedef UserVA32 UserVA32Const
- typedef UserVA64 UserVA64Const
- typedef void * UserVA
- typedef int64 PollDevHandle
- typedef uint16 utf16_t
- typedef uint32 MX_Rank

4.4.1 Define Documentation

4.4.1.1 `#define INCLUDE_ALLOW_USERLEVEL`

4.4.1.2 `#define INCLUDE_ALLOW_MODULE`

4.4.1.3 `#define INCLUDE_ALLOW_VMMON`

4.4.1.4 `#define INCLUDE_ALLOW_VMKERNEL`

4.4.1.5 `#define INCLUDE_ALLOW_VMKDRIVERS`

4.4.1.6 `#define INCLUDE_ALLOW_VMK_MODULE`

4.4.1.7 `#define INCLUDE_ALLOW_DISTRIBUTE`

4.4.1.8 `#define INCLUDE_ALLOW_VMCORE`

4.4.1.9 `#define vm_x86_64 0`

4.4.1.10 `#define vm_arm_64 0`

4.4.1.11 `#define vm_64bit (sizeof (void *) == 8)`

4.4.1.12 `#define _XTYPEDEF_BOOL`

4.4.1.13 `#define FALSE 0`

4.4.1.14 `#define TRUE 1`

4.4.1.15 `#define IS_BOOL(x) (((x) & ~1) == 0)`

4.4.1.16 `#define CONST3264(a) (a)`

4.4.1.17 `#define CONST3264U(a) (a)`

4.4.1.18 `#define MIN_INT8 ((int8)0x80)`

4.4.1.19 `#define MAX_INT8 ((int8)0x7f)`

4.4.1.20 `#define MIN_UINT8 ((uint8)0)`

4.4.1.21 `#define MAX_UINT8 ((uint8)0xff)`

4.4.1.22 `#define MIN_INT16 ((int16)0x8000)`

4.4.1.23 `#define MAX_INT16 ((int16)0x7fff)`

4.4.1.24 `#define MIN_UINT16 ((uint16)0)`

4.4.1.25 `#define MAX_UINT16 ((uint16)0xffff)`

4.4.1.26 `#define MIN_INT32 ((int32)0x80000000)`

4.4.1.27 `#define MAX_INT32 ((int32)0x7fffffff)`

4.4.1.28 `#define MIN_UINT32 ((uint32)0)`

4.4.1.29 `#define MAX_UINT32 ((uint32)0xffffffff)`

4.4.1.30 `#define MIN_INT64 (CONST64(0x8000000000000000))`

Index

- [_XTYPEDEF_BOOL](#)
[public_vm_basic_types.h](#), [63](#)
- [adapterType](#)
 - [VixDiskLibCreateParams](#), [12](#)
 - [VixDiskLibInfo](#), [15](#)
- [ALIGNED](#)
 - [public_vm_basic_types.h](#), [63](#)
- [allocatedBytes](#)
 - [VixDiskLibNasPluginStatXResult](#), [25](#)
- [allocType](#)
 - [VixDiskLibNasPluginStatXResult](#), [25](#)
- [APIC_INVALID_PPN](#)
 - [public_vm_basic_types.h](#), [63](#)
- [AsPercent](#)
 - [public_vm_basic_types.h](#), [63](#)
- [BA](#)
 - [public_vm_basic_types.h](#), [63](#)
- [biosGeo](#)
 - [VixDiskLibInfo](#), [15](#)
- [blocks](#)
 - [VixDiskLibBlockList](#), [6](#)
- [Bool](#)
 - [public_vm_basic_types.h](#), [63](#)
- [bottom](#)
 - [VMRect](#), [30](#)
- [BPN](#)
 - [public_vm_basic_types.h](#), [63](#)
- [BPN_2_UINT64](#)
 - [public_vm_basic_types.h](#), [63](#)
- [callback](#)
 - [VixDiskLibNasPluginProgressRecord](#), [20](#)
- [capacity](#)
 - [VixDiskLibCreateParams](#), [12](#)
 - [VixDiskLibInfo](#), [15](#)
- [cloneFlags](#)
 - [VixDiskLibNasPluginCloneFileParams](#), [17](#)
- [COLD](#)
 - [public_vm_basic_types.h](#), [63](#)
- [common](#)
 - [VixDiskLibNasPluginCloneFileParams](#), [17](#)
 - [VixDiskLibNasPluginResvSpaceParams](#), [22](#)
 - [VixDiskLibNasPluginStatXParams](#), [24](#)
 - [VixDiskLibNasPluginStatXResult](#), [25](#)
- [CONST](#)
 - [public_vm_basic_types.h](#), [63](#)
- [CONST3264](#)
 - [public_vm_basic_types.h](#), [63](#)
- [CONST3264U](#)
 - [public_vm_basic_types.h](#), [63](#)
- [CONST_FUNCTION](#)
 - [public_vm_basic_types.h](#), [63](#)
- [cookie](#)
 - [VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibSessionIdCreds](#), [10](#)
- [creds](#)
 - [VixDiskLibConnectParams](#), [8](#)
- [credType](#)
 - [VixDiskLibConnectParams](#), [8](#)
- [cylinders](#)
 - [VixDiskLibGeometry](#), [14](#)
- [datastoreMoRef](#)
 - [VixDiskLibDatastoreSpec](#), [13](#)
 - [VixDiskLibVStorageObjectSpec](#), [28](#)
- [diskLibPlugin](#)
 - [VixDiskLibNasPlugin](#), [16](#)
- [diskType](#)
 - [VixDiskLibCreateParams](#), [12](#)
- [distribute_vixDiskLib.h](#)
 - [VIXDISKLIB_ADAPTER_IDE](#), [37](#)
 - [VIXDISKLIB_ADAPTER_SCSI_-BUSLOGIC](#), [37](#)
 - [VIXDISKLIB_ADAPTER_SCSI_LSILOGIC](#), [37](#)
 - [VIXDISKLIB_ADAPTER_UNKNOWN](#), [37](#)
 - [VIXDISKLIB_CRED_SESSIONID](#), [37](#)
 - [VIXDISKLIB_CRED_SSPI](#), [37](#)
 - [VIXDISKLIB_CRED_TICKETID](#), [37](#)
 - [VIXDISKLIB_CRED_UID](#), [37](#)
 - [VIXDISKLIB_CRED_UNKNOWN](#), [37](#)
 - [VIXDISKLIB_DISK_MONOLITHIC_FLAT](#), [36](#)
 - [VIXDISKLIB_DISK_MONOLITHIC_-SPARSE](#), [36](#)
 - [VIXDISKLIB_DISK_SPLIT_FLAT](#), [37](#)
 - [VIXDISKLIB_DISK_SPLIT_SPARSE](#), [37](#)

- VIXDISKLIB_DISK_STREAM_-
OPTIMIZED, 37
- VIXDISKLIB_DISK_UNKNOWN, 37
- VIXDISKLIB_DISK_VMFS_FLAT, 37
- VIXDISKLIB_DISK_VMFS_SPARSE, 37
- VIXDISKLIB_DISK_VMFS_THIN, 37
- VIXDISKLIB_SPEC_DATASTORE, 37
- VIXDISKLIB_SPEC_UNKNOWN, 37
- VIXDISKLIB_SPEC_VMX, 37
- VIXDISKLIB_SPEC_VSTORAGE_-
OBJECT, 37
- distribute_vixDiskLibNasPlugin.h
 - VIXDISKLIB_NASPLUGIN_CLONEFILE_-
FLAG_DRYRUN, 55
 - VIXDISKLIB_NASPLUGIN_CLONEFILE_-
FLAG_GUARDED, 55
 - VIXDISKLIB_NASPLUGIN_CLONEFILE_-
FLAG_LAZY, 55
 - VIXDISKLIB_NASPLUGIN_CLONEFILE_-
FLAG_SKIPZEROES, 55
 - VIXDISKLIB_NASPLUGIN_CLONEFILE_-
FLAG_SRCDATASTORE_VALID,
55
 - VIXDISKLIB_NASPLUGIN_FILE_-
ALLOC_EZT, 54
 - VIXDISKLIB_NASPLUGIN_FILE_-
ALLOC_LZT, 54
 - VIXDISKLIB_NASPLUGIN_FILE_-
ALLOC_THIN, 54
 - VIXDISKLIB_NASPLUGIN_FILE_-
ALLOC_UNKNOWN, 54
 - VIXDISKLIB_NASPLUGIN_PRIM_-
CLONEFILE, 54
 - VIXDISKLIB_NASPLUGIN_PRIM_-
INVALID, 54
 - VIXDISKLIB_NASPLUGIN_PRIM_-
RESVSPACE, 54
 - VIXDISKLIB_NASPLUGIN_PRIM_STATX,
54
- distribute_vixDiskLibPlugin.h
 - VIXDISKLIB_PLUGIN_TYPE_NAS, 57
 - VIXDISKLIB_PLUGIN_TYPE_-
TRANSPORT, 57
 - VIXDISKLIB_PLUGIN_TYPE_-
TRANSPORT_NO_UNLOAD, 57
- distribute_vixDiskLib.h, 31
 - VixDiskLib_AllocateConnectParams, 50
 - VixDiskLib_Attach, 49
 - VixDiskLib_CheckRepair, 49
 - VixDiskLib_Cleanup, 38
 - VixDiskLib_Clone, 47
 - VixDiskLib_Close, 43
 - VixDiskLib_Connect, 39
 - VixDiskLib_ConnectEx, 39
 - VixDiskLib_Create, 41
 - VixDiskLib_CreateChild, 41
 - VixDiskLib_Defragment, 47
 - VixDiskLib_Disconnect, 40
 - VixDiskLib_EndAccess, 40
 - VixDiskLib_Exit, 38
 - VIXDISKLIB_FLAG_OPEN_-
COMPRESSION_FASTLZ, 36
 - VIXDISKLIB_FLAG_OPEN_-
COMPRESSION_MASK, 36
 - VIXDISKLIB_FLAG_OPEN_-
COMPRESSION_SKIPZ, 36
 - VIXDISKLIB_FLAG_OPEN_-
COMPRESSION_ZLIB, 36
 - VIXDISKLIB_FLAG_OPEN_READ_ONLY,
36
 - VIXDISKLIB_FLAG_OPEN_SINGLE_-
LINK, 36
 - VIXDISKLIB_FLAG_OPEN_-
UNBUFFERED, 36
 - VixDiskLib_Flush, 44
 - VixDiskLib_FreeBlockList, 42
 - VixDiskLib_FreeConnectParams, 50
 - VixDiskLib_FreeErrorText, 48
 - VixDiskLib_FreeInfo, 42
 - VixDiskLib_GetConnectParams, 50
 - VixDiskLib_GetErrorText, 48
 - VixDiskLib_GetInfo, 42
 - VixDiskLib_GetMetadataKeys, 46
 - VixDiskLib_GetTransportMode, 43
 - VixDiskLib_Grow, 46
 - VIXDISKLIB_HWVERSION_CURRENT,
36
 - VIXDISKLIB_HWVERSION_ESX30, 36
 - VIXDISKLIB_HWVERSION_ESX4X, 36
 - VIXDISKLIB_HWVERSION_ESX50, 36
 - VIXDISKLIB_HWVERSION_ESX51, 36
 - VIXDISKLIB_HWVERSION_ESX55, 36
 - VIXDISKLIB_HWVERSION_ESX60, 36
 - VIXDISKLIB_HWVERSION_ESX65, 36
 - VIXDISKLIB_HWVERSION_-
WORKSTATION_4, 36
 - VIXDISKLIB_HWVERSION_-
WORKSTATION_5, 36
 - VIXDISKLIB_HWVERSION_-
WORKSTATION_6, 36
 - VixDiskLib_Init, 38
 - VixDiskLib_InitEx, 37
 - VixDiskLib_IsAttachPossible, 48
 - VixDiskLib_ListTransportModes, 38
 - VIXDISKLIB_MAX_CHUNK_NUMBER,
36
 - VIXDISKLIB_MAX_CHUNK_SIZE, 36
 - VIXDISKLIB_MIN_CHUNK_SIZE, 36

- VixDiskLib_Open, 41
- VixDiskLib_PrepareForAccess, 39
- VixDiskLib_QueryAllocatedBlocks, 42
- VixDiskLib_Read, 43
- VixDiskLib_ReadAsync, 43
- VixDiskLib_ReadMetadata, 45
- VixDiskLib_Rename, 47
- VIXDISKLIB_SECTOR_SIZE, 36
- VixDiskLib_Shrink, 47
- VixDiskLib_SpaceNeededForClone, 49
- VixDiskLib_Unlink, 46
- VixDiskLib_Wait, 45
- VixDiskLib_Write, 44
- VixDiskLib_WriteAsync, 44
- VixDiskLib_WriteMetadata, 45
- VixDiskLibAdapterType, 37
- VixDiskLibCompletionCB, 36
- VixDiskLibConnection, 36
- VixDiskLibConnectParamsState, 36
- VixDiskLibCredType, 37
- VixDiskLibDiskType, 36
- VixDiskLibGenericLogFunc, 36
- VixDiskLibGenericLogVFunc, 36
- VixDiskLibHandle, 36
- VixDiskLibHandleStruct, 36
- VixDiskLibProgressFunc, 36
- VixDiskLibSectorType, 36
- VixDiskLibSpecType, 37
- distribute_vixDiskLibNasPlugin.h, 51
 - VIXDISKLIB_NASPLUGIN_FSTYPE_NFS, 52
 - VIXDISKLIB_NASPLUGIN_FSTYPE_NFS41, 52
 - VIXDISKLIB_NASPLUGIN_FSTYPE_VMFS, 52
 - VIXDISKLIB_NASPLUGIN_FSTYPE_VMFSL, 52
 - VIXDISKLIB_NASPLUGIN_INVALID_SESSION_ID, 52
 - VIXDISKLIB_NASPLUGIN_MAJOR_VERSION, 52
 - VIXDISKLIB_NASPLUGIN_MINOR_VERSION, 52
 - VixDiskLibNasPluginAllocType, 54
 - VixDiskLibNasPluginCloneFileFlags, 54
 - VixDiskLibNasPluginEndSession, 53
 - VixDiskLibNasPluginExecutePrimitive, 53
 - VixDiskLibNasPluginPeriodicCallback, 52
 - VixDiskLibNasPluginPrimitiveID, 54
 - VixDiskLibNasPluginSessionID, 52
 - VixDiskLibNasPluginStartSession, 52
 - VixDiskLibNasPluginSupportStatus, 53
- distribute_vixDiskLibPlugin.h, 56
 - VIXDISKLIB_PLUGIN_MAJOR_VERSION, 56
 - VIXDISKLIB_PLUGIN_MINOR_VERSION, 56
 - VixDiskLibPlugin_EntryPoint, 57
 - VixDiskLibPluginDone, 56
 - VixDiskLibPluginInit, 56
 - VixDiskLibPluginType, 57
- Done
 - VixDiskLibPlugin, 26
- dsSpec
 - VixDiskLibConnectParams, 8
 - VixDiskLibSpec, 27
- dstFileName
 - VixDiskLibNasPluginCloneFileParams, 17
- EndSession
 - VixDiskLibNasPlugin, 16
- ExecPrimitive
 - VixDiskLibNasPlugin, 16
- EXTERN
 - public_vm_basic_types.h, 63
- FALSE
 - public_vm_basic_types.h, 63
- fileName
 - VixDiskLibNasPluginResvSpaceParams, 22
 - VixDiskLibNasPluginStatXParams, 24
- FMTLA
 - public_vm_basic_types.h, 63
- FMTMODE
 - public_vm_basic_types.h, 63
- FMTPID
 - public_vm_basic_types.h, 63
- FMTUID
 - public_vm_basic_types.h, 63
- FMTVA
 - public_vm_basic_types.h, 63
- FMTVPN
 - public_vm_basic_types.h, 63
- fsType
 - VixDiskLibNasPluginDataStoreParams, 19
- fsVersion
 - VixDiskLibNasPluginDataStoreParams, 19
- heads
 - VixDiskLibGeometry, 14
- HOT
 - public_vm_basic_types.h, 63
- hwVersion
 - VixDiskLibCreateParams, 12
- id
 - VixDiskLibVStorageObjectSpec, 28

- INCLUDE_ALLOW_DISTRIBUTE
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_MODULE
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_USERLEVEL
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_VMCORE
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_VMK_MODULE
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_VMKDRIVERS
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_VMKERNEL
 - public_vm_basic_types.h, 63
- INCLUDE_ALLOW_VMMON
 - public_vm_basic_types.h, 63
- INFINITE_LOOP
 - public_vm_basic_types.h, 63
- Init
 - VixDiskLibPlugin, 26
- INLINE
 - public_vm_basic_types.h, 63
- INLINE_ALWAYS
 - public_vm_basic_types.h, 63
- INLINE_SINGLE_CALLER
 - public_vm_basic_types.h, 63
- int16
 - public_vm_basic_types.h, 63
- int32
 - public_vm_basic_types.h, 63
- int64
 - public_vm_basic_types.h, 63
- int8
 - public_vm_basic_types.h, 63
- INVALID_BPN
 - public_vm_basic_types.h, 63
- INVALID_CARTEL_ID
 - public_vm_basic_types.h, 63
- INVALID_CARTELGROUP_ID
 - public_vm_basic_types.h, 63
- INVALID_IOPN
 - public_vm_basic_types.h, 63
- INVALID_LPN
 - public_vm_basic_types.h, 63
- INVALID_LPN64
 - public_vm_basic_types.h, 63
- INVALID_MPN
 - public_vm_basic_types.h, 63
- INVALID_PAGENUM
 - public_vm_basic_types.h, 63
- INVALID_PAGENUM32
 - public_vm_basic_types.h, 63
- INVALID_PPN
 - public_vm_basic_types.h, 63
- INVALID_PPN32
 - public_vm_basic_types.h, 63
- INVALID_SESSION_ID
 - public_vm_basic_types.h, 63
- INVALID_VPN
 - public_vm_basic_types.h, 63
- INVALID_WORLD_ID
 - public_vm_basic_types.h, 63
- INVALID_WORLDLET_ID
 - public_vm_basic_types.h, 63
- IOA
 - public_vm_basic_types.h, 63
- IoHandle
 - public_vm_basic_types.h, 63
- IOPN
 - public_vm_basic_types.h, 63
- IS_BOOL
 - public_vm_basic_types.h, 63
- key
 - VixDiskLibConnect-Params::VixDiskLibCreds::VixDiskLibSessionIdCreds, 10
- LA
 - public_vm_basic_types.h, 63
- LA32
 - public_vm_basic_types.h, 63
- LA64
 - public_vm_basic_types.h, 63
- LA_2_LPN
 - public_vm_basic_types.h, 63
- LAST_LPN
 - public_vm_basic_types.h, 63
- LAST_LPN32
 - public_vm_basic_types.h, 63
- LAST_LPN64
 - public_vm_basic_types.h, 63
- left
 - VMRect, 30
- length
 - VixDiskLibBlock, 5
- LIKELY
 - public_vm_basic_types.h, 63
- localMountPoint
 - VixDiskLibNasPluginDataStoreParams, 19
- logicalSectorSize
 - VixDiskLibCreateParams, 12
 - VixDiskLibInfo, 15
- LPN
 - public_vm_basic_types.h, 63
- LPN32
 - public_vm_basic_types.h, 63
- LPN64
 - public_vm_basic_types.h, 63

- public_vm_basic_types.h, 63
- LPN_2_LA
 - public_vm_basic_types.h, 63
- LPN_MASK
 - public_vm_basic_types.h, 63
- LPN_MASK32
 - public_vm_basic_types.h, 63
- LPN_MASK64
 - public_vm_basic_types.h, 63
- MA
 - public_vm_basic_types.h, 63
- MA64
 - public_vm_basic_types.h, 63
- majorVersion
 - VixDiskLibPlugin, 26
- MAX_INT16
 - public_vm_basic_types.h, 63
- MAX_INT32
 - public_vm_basic_types.h, 63
- MAX_INT64
 - public_vm_basic_types.h, 63
- MAX_INT8
 - public_vm_basic_types.h, 63
- MAX_IOPN
 - public_vm_basic_types.h, 63
- MAX_MPN
 - public_vm_basic_types.h, 63
- MAX_PPN
 - public_vm_basic_types.h, 63
- MAX_PPN_BITS
 - public_vm_basic_types.h, 63
- MAX_UINT16
 - public_vm_basic_types.h, 63
- MAX_UINT32
 - public_vm_basic_types.h, 63
- MAX_UINT64
 - public_vm_basic_types.h, 63
- MAX_UINT8
 - public_vm_basic_types.h, 63
- MemHandle
 - public_vm_basic_types.h, 63
- MEMREF_MPN
 - public_vm_basic_types.h, 63
- MIN_INT16
 - public_vm_basic_types.h, 63
- MIN_INT32
 - public_vm_basic_types.h, 63
- MIN_INT64
 - public_vm_basic_types.h, 63
- MIN_INT8
 - public_vm_basic_types.h, 63
- MIN_UINT16
 - public_vm_basic_types.h, 63
- MIN_UINT32
 - public_vm_basic_types.h, 63
- MIN_UINT64
 - public_vm_basic_types.h, 63
- MIN_UINT8
 - public_vm_basic_types.h, 63
- minorVersion
 - VixDiskLibPlugin, 26
- MPN
 - public_vm_basic_types.h, 63
- MPN32
 - public_vm_basic_types.h, 63
- MPN38_MASK
 - public_vm_basic_types.h, 63
- MUST_CHECK_RETURN
 - public_vm_basic_types.h, 63
- MX_Rank
 - public_vm_basic_types.h, 63
- name
 - VixDiskLibPlugin, 26
- nfcHostPort
 - VixDiskLibConnectParams, 8
- NORETURN
 - public_vm_basic_types.h, 63
- numBlocks
 - VixDiskLibBlockList, 6
- numLinks
 - VixDiskLibInfo, 15
- offset
 - VixDiskLibBlock, 5
- PA
 - public_vm_basic_types.h, 63
- PA32
 - public_vm_basic_types.h, 63
- PA64
 - public_vm_basic_types.h, 63
- PageCnt
 - public_vm_basic_types.h, 63
- PageNum
 - public_vm_basic_types.h, 63
- parentFileNameHint
 - VixDiskLibInfo, 15
- password
 - VixDiskLibConnect-Params::VixDiskLibCreds::VixDiskLibUidPasswdCreds, 11
- Percent
 - public_vm_basic_types.h, 63
- physGeo
 - VixDiskLibInfo, 15
- physicalSectorSize

- VixDiskLibCreateParams, 12
- VixDiskLibInfo, 15
- PhysMemOff
 - public_vm_basic_types.h, 63
- PhysMemSize
 - public_vm_basic_types.h, 63
- PollDevHandle
 - public_vm_basic_types.h, 63
- port
 - VixDiskLibConnectParams, 8
- PPN
 - public_vm_basic_types.h, 63
- PPN32
 - public_vm_basic_types.h, 63
- PPN64
 - public_vm_basic_types.h, 63
- primitiveID
 - VixDiskLibNasPluginCommonParams, 18
- PRINTF_DECL
 - public_vm_basic_types.h, 63
- private
 - VixDiskLibNasPluginProgressRecord, 20
- privateUse
 - VixDiskLibConnectParams, 8
- progressBytes
 - VixDiskLibNasPluginProgressRecord, 20
- progressRecord
 - VixDiskLibNasPluginCommonParams, 18
- public_vm_basic_types.h, 58
 - _XTYPEDEF_BOOL, 63
 - ALIGNED, 63
 - APIC_INVALID_PPN, 63
 - AsPercent, 63
 - BA, 63
 - Bool, 63
 - BPN, 63
 - BPN_2_UINT64, 63
 - COLD, 63
 - CONST, 63
 - CONST3264, 63
 - CONST3264U, 63
 - CONST_FUNCTION, 63
 - EXTERN, 63
 - FALSE, 63
 - FMTLA, 63
 - FMTMODE, 63
 - FMTPID, 63
 - FMTUID, 63
 - FMTVA, 63
 - FMTVPN, 63
 - HOT, 63
 - INCLUDE_ALLOW_DISTRIBUTE, 63
 - INCLUDE_ALLOW_MODULE, 63
 - INCLUDE_ALLOW_USERLEVEL, 63
 - INCLUDE_ALLOW_VMCORE, 63
 - INCLUDE_ALLOW_VMK_MODULE, 63
 - INCLUDE_ALLOW_VMKDRIVERS, 63
 - INCLUDE_ALLOW_VMKERNEL, 63
 - INCLUDE_ALLOW_VMMON, 63
 - INFINITE_LOOP, 63
 - INLINE, 63
 - INLINE_ALWAYS, 63
 - INLINE_SINGLE_CALLER, 63
 - int16, 63
 - int32, 63
 - int64, 63
 - int8, 63
 - INVALID_BPN, 63
 - INVALID_CARTEL_ID, 63
 - INVALID_CARTELGROUP_ID, 63
 - INVALID_IOPN, 63
 - INVALID_LPN, 63
 - INVALID_LPN64, 63
 - INVALID_MPN, 63
 - INVALID_PAGENUM, 63
 - INVALID_PAGENUM32, 63
 - INVALID_PPN, 63
 - INVALID_PPN32, 63
 - INVALID_SESSION_ID, 63
 - INVALID_VPN, 63
 - INVALID_WORLD_ID, 63
 - INVALID_WORLDLET_ID, 63
 - IOA, 63
 - IoHandle, 63
 - IOPN, 63
 - IS_BOOL, 63
 - LA, 63
 - LA32, 63
 - LA64, 63
 - LA_2_LPN, 63
 - LAST_LPN, 63
 - LAST_LPN32, 63
 - LAST_LPN64, 63
 - LIKELY, 63
 - LPN, 63
 - LPN32, 63
 - LPN64, 63
 - LPN_2_LA, 63
 - LPN_MASK, 63
 - LPN_MASK32, 63
 - LPN_MASK64, 63
 - MA, 63
 - MA64, 63
 - MAX_INT16, 63
 - MAX_INT32, 63
 - MAX_INT64, 63
 - MAX_INT8, 63
 - MAX_IOPN, 63

- MAX_MPN, 63
- MAX_PPN, 63
- MAX_PPN_BITS, 63
- MAX_UINT16, 63
- MAX_UINT32, 63
- MAX_UINT64, 63
- MAX_UINT8, 63
- MemHandle, 63
- MEMREF_MPN, 63
- MIN_INT16, 63
- MIN_INT32, 63
- MIN_INT64, 63
- MIN_INT8, 63
- MIN_UINT16, 63
- MIN_UINT32, 63
- MIN_UINT64, 63
- MIN_UINT8, 63
- MPN, 63
- MPN32, 63
- MPN38_MASK, 63
- MUST_CHECK_RETURN, 63
- MX_Rank, 63
- NORETURN, 63
- PA, 63
- PA32, 63
- PA64, 63
- PageCnt, 63
- PageNum, 63
- Percent, 63
- PhysMemOff, 63
- PhysMemSize, 63
- PollDevHandle, 63
- PPN, 63
- PPN32, 63
- PPN64, 63
- PRINTF_DECL, 63
- Reg16, 63
- Reg32, 63
- Reg64, 63
- Reg8, 63
- RELEASED_MPN, 63
- RESERVED_MPN, 63
- SCANF_DECL, 63
- SectorType, 63
- SIDE_EFFECT_FREE, 63
- TCA, 63
- TPA, 63
- TPPN, 63
- TPPN64, 63
- TRUE, 63
- uint16, 63
- uint32, 63
- uint64, 63
- UINT64_2_BPN, 63
- uint8, 63
- UNLIKELY, 63
- UNUSED_PARAM, 63
- UNUSED_TYPE, 63
- UNUSED_VARIABLE, 63
- UReg16, 63
- UReg32, 63
- UReg64, 63
- UReg8, 63
- User_CartelGroupID, 63
- User_CartelID, 63
- User_SessionID, 63
- UserVA, 63
- UserVA32, 63
- UserVA32Const, 63
- UserVA64, 63
- UserVA64Const, 63
- UserVAConst, 63
- utf16_t, 63
- VA, 63
- VA32, 63
- VA64, 63
- vm_64bit, 63
- vm_arm_64, 63
- vm_x86_64, 63
- VmTimeRealClock, 63
- VmTimeType, 63
- VmTimeVirtualClock, 63
- VMX86_EXTERN_DATA, 63
- VPN, 63
- VPN32, 63
- VPN64, 63
- World_ID, 63
- Worldlet_ID, 63
- Reg16
 - public_vm_basic_types.h, 63
- Reg32
 - public_vm_basic_types.h, 63
- Reg64
 - public_vm_basic_types.h, 63
- Reg8
 - public_vm_basic_types.h, 63
- RELEASED_MPN
 - public_vm_basic_types.h, 63
- remoteIP
 - VixDiskLibNasPluginDataStoreParams, 19
- remoteMountPoint
 - VixDiskLibNasPluginDataStoreParams, 19
- reserved
 - VixDiskLibConnectParams, 8
- RESERVED_MPN
 - public_vm_basic_types.h, 63
- result

- VixDiskLibNasPluginCommonParams, 18
- right
 - VMRect, 30
- SCANF_DECL
 - public_vm_basic_types.h, 63
- sectors
 - VixDiskLibGeometry, 14
- SectorType
 - public_vm_basic_types.h, 63
- serverName
 - VixDiskLibConnectParams, 8
- sessionId
 - VixDiskLibConnect-Params::VixDiskLibCreds, 9
- SIDE_EFFECT_FREE
 - public_vm_basic_types.h, 63
- spec
 - VixDiskLibConnectParams, 8
- specType
 - VixDiskLibConnectParams, 8
- srcDataStoreInfo
 - VixDiskLibNasPluginCloneFileParams, 17
- srcFileName
 - VixDiskLibNasPluginCloneFileParams, 17
- ssId
 - VixDiskLibVStorageObjectSpec, 28
- StartSession
 - VixDiskLibNasPlugin, 16
- state
 - VixDiskLibConnectParams, 8
- status
 - VixDiskLibNasPluginResultCommon, 21
- SupportStatus
 - VixDiskLibNasPlugin, 16
- TCA
 - public_vm_basic_types.h, 63
- thumbPrint
 - VixDiskLibConnectParams, 8
- ticketId
 - VixDiskLibConnect-Params::VixDiskLibCreds, 9
- timeoutMS
 - VixDiskLibNasPluginSessionParams, 23
- top
 - VMRect, 30
- totalBytes
 - VixDiskLibNasPluginStatXResult, 25
- TPA
 - public_vm_basic_types.h, 63
- TPPN
 - public_vm_basic_types.h, 63
- TPPN64
 - public_vm_basic_types.h, 63
- public_vm_basic_types.h, 63
- TRUE
 - public_vm_basic_types.h, 63
- type
 - VixDiskLibPlugin, 26
- uid
 - VixDiskLibConnect-Params::VixDiskLibCreds, 9
- uint16
 - public_vm_basic_types.h, 63
- uint32
 - public_vm_basic_types.h, 63
- uint64
 - public_vm_basic_types.h, 63
- UINT64_2_BPN
 - public_vm_basic_types.h, 63
- uint8
 - public_vm_basic_types.h, 63
- uniqueBytes
 - VixDiskLibNasPluginStatXResult, 25
- UNLIKELY
 - public_vm_basic_types.h, 63
- UNUSED_PARAM
 - public_vm_basic_types.h, 63
- UNUSED_TYPE
 - public_vm_basic_types.h, 63
- UNUSED_VARIABLE
 - public_vm_basic_types.h, 63
- updateBytes
 - VixDiskLibNasPluginProgressRecord, 20
- UReg16
 - public_vm_basic_types.h, 63
- UReg32
 - public_vm_basic_types.h, 63
- UReg64
 - public_vm_basic_types.h, 63
- UReg8
 - public_vm_basic_types.h, 63
- User_CartelGroupID
 - public_vm_basic_types.h, 63
- User_CartelID
 - public_vm_basic_types.h, 63
- User_SessionID
 - public_vm_basic_types.h, 63
- userName
 - VixDiskLibConnect-Params::VixDiskLibCreds::VixDiskLibSessionIdCreds, 10
 - VixDiskLibConnect-Params::VixDiskLibCreds::VixDiskLibUidPasswdCreds, 11
- UserVA
 - public_vm_basic_types.h, 63

- UserVA32
 - public_vm_basic_types.h, 63
- UserVA32Const
 - public_vm_basic_types.h, 63
- UserVA64
 - public_vm_basic_types.h, 63
- UserVA64Const
 - public_vm_basic_types.h, 63
- UserVAConst
 - public_vm_basic_types.h, 63
- utf16_t
 - public_vm_basic_types.h, 63
- uuid
 - VixDiskLibInfo, 15
- VA
 - public_vm_basic_types.h, 63
- VA32
 - public_vm_basic_types.h, 63
- VA64
 - public_vm_basic_types.h, 63
- vimApiVer
 - VixDiskLibConnectParams, 8
- VIXDISKLIB_ADAPTER_IDE
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_ADAPTER_SCSI_BUSLOGIC
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_ADAPTER_SCSI_LSILOGIC
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_ADAPTER_UNKNOWN
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_CRED_SESSIONID
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_CRED_SSPI
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_CRED_TICKETID
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_CRED_UID
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_CRED_UNKNOWN
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_MONOLITHIC_FLAT
 - distribute_vixDiskLib.h, 36
- VIXDISKLIB_DISK_MONOLITHIC_SPARSE
 - distribute_vixDiskLib.h, 36
- VIXDISKLIB_DISK_SPLIT_FLAT
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_SPLIT_SPARSE
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_STREAM_OPTIMIZED
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_UNKNOWN
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_VMFS_FLAT
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_VMFS_SPARSE
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_DISK_VMFS_THIN
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_NASPLUGIN_CLONEFILE_-
 - FLAG_DRYRUN
 - distribute_vixDiskLibNasPlugin.h, 55
 - FLAG_GUARDED
 - distribute_vixDiskLibNasPlugin.h, 55
 - FLAG_LAZY
 - distribute_vixDiskLibNasPlugin.h, 55
 - FLAG_SKIPZEROES
 - distribute_vixDiskLibNasPlugin.h, 55
 - FLAG_SRCDATASTORE_VALID
 - distribute_vixDiskLibNasPlugin.h, 55
- VIXDISKLIB_NASPLUGIN_FILE_ALLOC_EZT
 - distribute_vixDiskLibNasPlugin.h, 54
- VIXDISKLIB_NASPLUGIN_FILE_ALLOC_LZT
 - distribute_vixDiskLibNasPlugin.h, 54
- VIXDISKLIB_NASPLUGIN_FILE_ALLOC_-
 - THIN
 - distribute_vixDiskLibNasPlugin.h, 54
 - UNKNOWN
 - distribute_vixDiskLibNasPlugin.h, 54
- VIXDISKLIB_NASPLUGIN_PRIM_-
 - CLONEFILE
 - distribute_vixDiskLibNasPlugin.h, 54
 - PRIM_INVALID
 - distribute_vixDiskLibNasPlugin.h, 54
- VIXDISKLIB_NASPLUGIN_PRIM_-
 - RESVSPACE
 - distribute_vixDiskLibNasPlugin.h, 54
 - PRIM_STATX
 - distribute_vixDiskLibNasPlugin.h, 54
- VIXDISKLIB_PLUGIN_TYPE_NAS
 - distribute_vixDiskLibPlugin.h, 57
- VIXDISKLIB_PLUGIN_TYPE_TRANSPORT
 - distribute_vixDiskLibPlugin.h, 57
- VIXDISKLIB_PLUGIN_TYPE_TRANSPORT_-
 - NO_UNLOAD
 - distribute_vixDiskLibPlugin.h, 57
- VIXDISKLIB_SPEC_DATASTORE
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_SPEC_UNKNOWN
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_SPEC_VMX
 - distribute_vixDiskLib.h, 37
- VIXDISKLIB_SPEC_VSTORAGE_OBJECT

- [distribute_vixDiskLib.h](#), 37
- [VixDiskLib_AllocateConnectParams](#)
 - [distribute_vixDiskLib.h](#), 50
- [VixDiskLib_Attach](#)
 - [distribute_vixDiskLib.h](#), 49
- [VixDiskLib_CheckRepair](#)
 - [distribute_vixDiskLib.h](#), 49
- [VixDiskLib_Cleanup](#)
 - [distribute_vixDiskLib.h](#), 38
- [VixDiskLib_Clone](#)
 - [distribute_vixDiskLib.h](#), 47
- [VixDiskLib_Close](#)
 - [distribute_vixDiskLib.h](#), 43
- [VixDiskLib_Connect](#)
 - [distribute_vixDiskLib.h](#), 39
- [VixDiskLib_ConnectEx](#)
 - [distribute_vixDiskLib.h](#), 39
- [VixDiskLib_Create](#)
 - [distribute_vixDiskLib.h](#), 41
- [VixDiskLib_CreateChild](#)
 - [distribute_vixDiskLib.h](#), 41
- [VixDiskLib_Defragment](#)
 - [distribute_vixDiskLib.h](#), 47
- [VixDiskLib_Disconnect](#)
 - [distribute_vixDiskLib.h](#), 40
- [VixDiskLib_EndAccess](#)
 - [distribute_vixDiskLib.h](#), 40
- [VixDiskLib_Exit](#)
 - [distribute_vixDiskLib.h](#), 38
- [VIXDISKLIB_FLAG_OPEN_COMPRESSION_-FASTLZ](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_COMPRESSION_-MASK](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_COMPRESSION_-SKIPZ](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_COMPRESSION_-ZLIB](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_READ_ONLY](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_SINGLE_LINK](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_FLAG_OPEN_UNBUFFERED](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLib_Flush](#)
 - [distribute_vixDiskLib.h](#), 44
- [VixDiskLib_FreeBlockList](#)
 - [distribute_vixDiskLib.h](#), 42
- [VixDiskLib_FreeConnectParams](#)
 - [distribute_vixDiskLib.h](#), 50
- [VixDiskLib_FreeErrorText](#)
 - [distribute_vixDiskLib.h](#), 48
- [VixDiskLib_FreeInfo](#)
 - [distribute_vixDiskLib.h](#), 42
- [VixDiskLib_GetConnectParams](#)
 - [distribute_vixDiskLib.h](#), 50
- [VixDiskLib_GetErrorText](#)
 - [distribute_vixDiskLib.h](#), 48
- [VixDiskLib_GetInfo](#)
 - [distribute_vixDiskLib.h](#), 42
- [VixDiskLib_GetMetadataKeys](#)
 - [distribute_vixDiskLib.h](#), 46
- [VixDiskLib_GetTransportMode](#)
 - [distribute_vixDiskLib.h](#), 43
- [VixDiskLib_Grow](#)
 - [distribute_vixDiskLib.h](#), 46
- [VIXDISKLIB_HWVERSION_CURRENT](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX30](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX4X](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX50](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX51](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX55](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX60](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_ESX65](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_WORKSTATION_-4](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_WORKSTATION_-5](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_HWVERSION_WORKSTATION_-6](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLib_Init](#)
 - [distribute_vixDiskLib.h](#), 38
- [VixDiskLib_InitEx](#)
 - [distribute_vixDiskLib.h](#), 37
- [VixDiskLib_IsAttachPossible](#)
 - [distribute_vixDiskLib.h](#), 48
- [VixDiskLib_ListTransportModes](#)
 - [distribute_vixDiskLib.h](#), 38
- [VIXDISKLIB_MAX_CHUNK_NUMBER](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_MAX_CHUNK_SIZE](#)
 - [distribute_vixDiskLib.h](#), 36
- [VIXDISKLIB_MIN_CHUNK_SIZE](#)
 - [distribute_vixDiskLib.h](#), 36

- VIXDISKLIB_NASPLUGIN_FSTYPE_NFS
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_FSTYPE_NFS41
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_FSTYPE_VMFS
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_FSTYPE_VMFSL
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_INVALID_SESSION_ID
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_MAJOR_VERSION
 - distributed_vixDiskLibNasPlugin.h, 52
- VIXDISKLIB_NASPLUGIN_MINOR_VERSION
 - distributed_vixDiskLibNasPlugin.h, 52
- VixDiskLib_Open
 - distributed_vixDiskLib.h, 41
- VIXDISKLIB_PLUGIN_MAJOR_VERSION
 - distributed_vixDiskLibPlugin.h, 56
- VIXDISKLIB_PLUGIN_MINOR_VERSION
 - distributed_vixDiskLibPlugin.h, 56
- VixDiskLib_PrepareForAccess
 - distributed_vixDiskLib.h, 39
- VixDiskLib_QueryAllocatedBlocks
 - distributed_vixDiskLib.h, 42
- VixDiskLib_Read
 - distributed_vixDiskLib.h, 43
- VixDiskLib_ReadAsync
 - distributed_vixDiskLib.h, 43
- VixDiskLib_ReadMetadata
 - distributed_vixDiskLib.h, 45
- VixDiskLib_Rename
 - distributed_vixDiskLib.h, 47
- VIXDISKLIB_SECTOR_SIZE
 - distributed_vixDiskLib.h, 36
- VixDiskLib_Shrink
 - distributed_vixDiskLib.h, 47
- VixDiskLib_SpaceNeededForClone
 - distributed_vixDiskLib.h, 49
- VixDiskLib_Unlink
 - distributed_vixDiskLib.h, 46
- VixDiskLib_Wait
 - distributed_vixDiskLib.h, 45
- VixDiskLib_Write
 - distributed_vixDiskLib.h, 44
- VixDiskLib_WriteAsync
 - distributed_vixDiskLib.h, 44
- VixDiskLib_WriteMetadata
 - distributed_vixDiskLib.h, 45
- VixDiskLibAdapterType
 - distributed_vixDiskLib.h, 37
- VixDiskLibBlock, 5
 - length, 5
 - offset, 5
- VixDiskLibBlockList, 6
 - blocks, 6
 - numBlocks, 6
- VixDiskLibCompletionCB
 - distributed_vixDiskLib.h, 36
- VixDiskLibConnection
 - distributed_vixDiskLib.h, 36
- VixDiskLibConnectParams, 7
 - creds, 8
 - credType, 8
 - dsSpec, 8
 - nfcHostPort, 8
 - port, 8
 - privateUse, 8
 - reserved, 8
 - serverName, 8
 - spec, 8
 - specType, 8
 - state, 8
 - thumbPrint, 8
 - vimApiVer, 8
 - vmxSpec, 8
 - vStorageObjSpec, 8
- VixDiskLibConnectParams::VixDiskLibCreds, 9
 - sessionId, 9
 - ticketId, 9
 - uid, 9
- VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibSessionIdCreds, 10
 - cookie, 10
 - key, 10
 - userName, 10
- VixDiskLibConnectParams::VixDiskLibCreds::VixDiskLibUidPasswdCreds, 11
 - password, 11
 - userName, 11
- VixDiskLibConnectParamsState
 - distributed_vixDiskLib.h, 36
- VixDiskLibCreateParams, 12
 - adapterType, 12
 - capacity, 12
 - diskType, 12
 - hwVersion, 12
 - logicalSectorSize, 12
 - physicalSectorSize, 12
- VixDiskLibCredType
 - distributed_vixDiskLib.h, 37
- VixDiskLibDatastoreSpec, 13
 - datastoreMoRef, 13
- VixDiskLibDiskType
 - distributed_vixDiskLib.h, 36
- VixDiskLibGenericLogFunc
 - distributed_vixDiskLib.h, 36
- VixDiskLibGenericLogVFunc

- [distribute_vixDiskLib.h](#), 36
- [VixDiskLibGeometry](#), 14
 - [cylinders](#), 14
 - [heads](#), 14
 - [sectors](#), 14
- [VixDiskLibHandle](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLibHandleStruct](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLibInfo](#), 15
 - [adapterType](#), 15
 - [biosGeo](#), 15
 - [capacity](#), 15
 - [logicalSectorSize](#), 15
 - [numLinks](#), 15
 - [parentFileNameHint](#), 15
 - [physGeo](#), 15
 - [physicalSectorSize](#), 15
 - [uuid](#), 15
- [VixDiskLibNasPlugin](#), 16
 - [diskLibPlugin](#), 16
 - [EndSession](#), 16
 - [ExecPrimitive](#), 16
 - [StartSession](#), 16
 - [SupportStatus](#), 16
- [VixDiskLibNasPluginAllocType](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 54
- [VixDiskLibNasPluginCloneFileFlags](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 54
- [VixDiskLibNasPluginCloneFileParams](#), 17
 - [cloneFlags](#), 17
 - [common](#), 17
 - [dstFileName](#), 17
 - [srcDataStoreInfo](#), 17
 - [srcFileName](#), 17
- [VixDiskLibNasPluginCommonParams](#), 18
 - [primitiveID](#), 18
 - [progressRecord](#), 18
 - [result](#), 18
- [VixDiskLibNasPluginDataStoreParams](#), 19
 - [fsType](#), 19
 - [fsVersion](#), 19
 - [localMountPoint](#), 19
 - [remoteIP](#), 19
 - [remoteMountPoint](#), 19
- [VixDiskLibNasPluginEndSession](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 53
- [VixDiskLibNasPluginExecutePrimitive](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 53
- [VixDiskLibNasPluginPeriodicCallback](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 52
- [VixDiskLibNasPluginPrimitiveID](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 54
- [VixDiskLibNasPluginProgressRecord](#), 20
 - [callback](#), 20
 - [private](#), 20
 - [progressBytes](#), 20
 - [updateBytes](#), 20
- [VixDiskLibNasPluginResultCommon](#), 21
 - [status](#), 21
- [VixDiskLibNasPluginResvSpaceParams](#), 22
 - [common](#), 22
 - [fileName](#), 22
- [VixDiskLibNasPluginSessionID](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 52
- [VixDiskLibNasPluginSessionParams](#), 23
 - [timeoutMS](#), 23
- [VixDiskLibNasPluginStartSession](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 52
- [VixDiskLibNasPluginStatXParams](#), 24
 - [common](#), 24
 - [fileName](#), 24
- [VixDiskLibNasPluginStatXResult](#), 25
 - [allocedBytes](#), 25
 - [allocType](#), 25
 - [common](#), 25
 - [totalBytes](#), 25
 - [uniqueBytes](#), 25
- [VixDiskLibNasPluginSupportStatus](#)
 - [distribute_vixDiskLibNasPlugin.h](#), 53
- [VixDiskLibPlugin](#), 26
 - [Done](#), 26
 - [Init](#), 26
 - [majorVersion](#), 26
 - [minorVersion](#), 26
 - [name](#), 26
 - [type](#), 26
- [VixDiskLibPlugin_EntryPoint](#)
 - [distribute_vixDiskLibPlugin.h](#), 57
- [VixDiskLibPluginDone](#)
 - [distribute_vixDiskLibPlugin.h](#), 56
- [VixDiskLibPluginInit](#)
 - [distribute_vixDiskLibPlugin.h](#), 56
- [VixDiskLibPluginType](#)
 - [distribute_vixDiskLibPlugin.h](#), 57
- [VixDiskLibProgressFunc](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLibSectorType](#)
 - [distribute_vixDiskLib.h](#), 36
- [VixDiskLibSpec](#), 27
 - [dsSpec](#), 27
 - [vmxSpec](#), 27
 - [vStorageObjSpec](#), 27
- [VixDiskLibSpecType](#)
 - [distribute_vixDiskLib.h](#), 37
- [VixDiskLibVStorageObjectSpec](#), 28
 - [datastoreMoRef](#), 28
 - [id](#), 28

- ssId, [28](#)
- vm_64bit
 - [public_vm_basic_types.h](#), [63](#)
- vm_arm_64
 - [public_vm_basic_types.h](#), [63](#)
- vm_x86_64
 - [public_vm_basic_types.h](#), [63](#)
- VMPPoint, [29](#)
 - x, [29](#)
 - y, [29](#)
- VMRect, [30](#)
 - bottom, [30](#)
 - left, [30](#)
 - right, [30](#)
 - top, [30](#)
- VmTimeRealClock
 - [public_vm_basic_types.h](#), [63](#)
- VmTimeType
 - [public_vm_basic_types.h](#), [63](#)
- VmTimeVirtualClock
 - [public_vm_basic_types.h](#), [63](#)
- VMX86_EXTERN_DATA
 - [public_vm_basic_types.h](#), [63](#)
- vmxSpec
 - VixDiskLibConnectParams, [8](#)
 - VixDiskLibSpec, [27](#)
- VPN
 - [public_vm_basic_types.h](#), [63](#)
- VPN32
 - [public_vm_basic_types.h](#), [63](#)
- VPN64
 - [public_vm_basic_types.h](#), [63](#)
- vStorageObjSpec
 - VixDiskLibConnectParams, [8](#)
 - VixDiskLibSpec, [27](#)
- World_ID
 - [public_vm_basic_types.h](#), [63](#)
- Worldlet_ID
 - [public_vm_basic_types.h](#), [63](#)
- x
 - VMPPoint, [29](#)
- y
 - VMPPoint, [29](#)