

**cmpi-router**  
1.0.0

Generated by Doxygen 1.6.1

Tue Sep 22 18:42:12 2009



# Contents

<b>1</b>	<b>Todo List</b>	<b>1</b>
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Data Structure Documentation</b>	<b>7</b>
4.1	LANEndpoint Struct Reference . . . . .	7
4.1.1	Detailed Description . . . . .	8
4.1.2	Field Documentation . . . . .	8
4.1.2.1	aliasAddresses . . . . .	8
4.1.2.2	availReqStates . . . . .	8
4.1.2.3	caption . . . . .	8
4.1.2.4	communicationStatus . . . . .	8
4.1.2.5	creationClassName . . . . .	8
4.1.2.6	description . . . . .	9
4.1.2.7	detailedStatus . . . . .	9
4.1.2.8	elementName . . . . .	9
4.1.2.9	enabledDefault . . . . .	9
4.1.2.10	enabledState . . . . .	9
4.1.2.11	groupAddresses . . . . .	9
4.1.2.12	healthState . . . . .	9
4.1.2.13	instanceID . . . . .	9
4.1.2.14	lanID . . . . .	9
4.1.2.15	macAddress . . . . .	10
4.1.2.16	maxDataSize . . . . .	10
4.1.2.17	name . . . . .	10

4.1.2.18	nameFormat	10
4.1.2.19	operatingStatus	10
4.1.2.20	operationalStatus	10
4.1.2.21	otherEnabledState	10
4.1.2.22	otherTypeDescription	10
4.1.2.23	primaryStatus	10
4.1.2.24	protocolIFType	11
4.1.2.25	requestedState	11
4.1.2.26	statusDescriptions	11
4.1.2.27	systemCreationClassName	11
4.1.2.28	systemName	11
4.1.2.29	transitioningToState	11
4.2	LANEndpointList Struct Reference	12
4.2.1	Detailed Description	12
4.2.2	Field Documentation	12
4.2.2.1	next	12
4.2.2.2	sptr	12
4.3	nextHopIP Struct Reference	13
4.3.1	Detailed Description	13
4.3.2	Field Documentation	13
4.3.2.1	addressType	13
4.3.2.2	adminDistance	13
4.3.2.3	caption	14
4.3.2.4	description	14
4.3.2.5	dstAddress	14
4.3.2.6	dstMask	14
4.3.2.7	elementName	14
4.3.2.8	instanceID	14
4.3.2.9	isStatic	14
4.3.2.10	otherDerivation	14
4.3.2.11	prefixLength	14
4.3.2.12	routeDerivation	15
4.3.2.13	routeGateway	15
4.3.2.14	routeMetric	15
4.3.2.15	routeOutputIf	15
4.3.2.16	routeScope	15

4.3.2.17	routeTable	15
4.3.2.18	routeType	15
4.3.2.19	typeOfRoute	15
4.4	nextHopIPLIST Struct Reference	16
4.4.1	Detailed Description	16
4.4.2	Field Documentation	16
4.4.2.1	next	16
4.4.2.2	sptr	16
4.5	nlLinkInfo Struct Reference	17
4.5.1	Detailed Description	17
4.5.2	Field Documentation	17
4.5.2.1	address	17
4.5.2.2	addressLen	17
4.5.2.3	broadcast	18
4.5.2.4	broadcastLen	18
4.5.2.5	change	18
4.5.2.6	family	18
4.5.2.7	flags	18
4.5.2.8	ifname	18
4.5.2.9	index	18
4.5.2.10	link	18
4.5.2.11	linkmode	18
4.5.2.12	map	19
4.5.2.13	mtu	19
4.5.2.14	operstate	19
4.5.2.15	qdisc	19
4.5.2.16	stats	19
4.5.2.17	txqlen	19
4.5.2.18	type	19
4.6	nlLinkInfoList Struct Reference	20
4.6.1	Detailed Description	20
4.6.2	Field Documentation	20
4.6.2.1	next	20
4.6.2.2	sptr	20
4.7	nlRouteInfo Struct Reference	21
4.7.1	Detailed Description	21

4.7.2	Field Documentation	21
4.7.2.1	dstAddr	21
4.7.2.2	dstLen	21
4.7.2.3	family	22
4.7.2.4	gw	22
4.7.2.5	inputIf	22
4.7.2.6	metrics	22
4.7.2.7	outputIf	22
4.7.2.8	prefSrc	22
4.7.2.9	priority	22
4.7.2.10	protocol	22
4.7.2.11	scope	22
4.7.2.12	srcAddr	23
4.7.2.13	srcLen	23
4.7.2.14	table	23
4.7.2.15	tos	23
4.7.2.16	type	23
4.8	nlRouteInfoList Struct Reference	24
4.8.1	Detailed Description	24
4.8.2	Field Documentation	24
4.8.2.1	next	24
4.8.2.2	sptr	24
4.9	nlSockHandle Struct Reference	25
4.9.1	Detailed Description	25
4.9.2	Field Documentation	25
4.9.2.1	fd	25
4.9.2.2	local	25
<b>5</b>	<b>File Documentation</b>	<b>27</b>
5.1	/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_LANEndpoint.h File Reference	27
5.1.1	Detailed Description	27
5.1.2	Function Documentation	28
5.1.2.1	_makeInst_LANEndpoint	28
5.1.2.2	_makePath_LANEndpoint	28
5.1.3	Variable Documentation	29
5.1.3.1	_ClassName	29
5.2	/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_NextHopIPRoute.h File Reference	30

5.2.1	Detailed Description	30
5.2.2	Function Documentation	30
5.2.2.1	<a href="#">_makeInst_NextHopIPRoute</a>	31
5.2.2.2	<a href="#">_makePath_NextHopIPRoute</a>	31
5.2.3	Variable Documentation	31
5.2.3.1	<a href="#">_ClassName</a>	31
5.3	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_RouteUsesEndpoint.h File Reference</a>	32
5.3.1	Detailed Description	32
5.3.2	Enumeration Type Documentation	33
5.3.2.1	<a href="#">"@0</a>	33
5.3.3	Function Documentation	33
5.3.3.1	<a href="#">_assoc_get_NextHopRoute_insts</a>	33
5.3.3.2	<a href="#">_assoc_get_ProtocolEndpoint_insts</a>	34
5.3.3.3	<a href="#">_assoc_RouteUsesEndpoint</a>	34
5.3.3.4	<a href="#">_makeInst_RouteUsesEndpoint</a>	35
5.3.3.5	<a href="#">_makePath_RouteUsesEndpoint</a>	35
5.3.4	Variable Documentation	35
5.3.4.1	<a href="#">_ClassName</a>	35
5.3.4.2	<a href="#">_RefLeft</a>	35
5.3.4.3	<a href="#">_RefLeftClass</a>	35
5.3.4.4	<a href="#">_RefLeftClasses</a>	36
5.3.4.5	<a href="#">_RefRight</a>	36
5.3.4.6	<a href="#">_RefRightClass</a>	36
5.3.4.7	<a href="#">_RefRightClasses</a>	36
5.4	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_LANEndpoint.h File Reference</a>	37
5.4.1	Detailed Description	38
5.4.2	Define Documentation	39
5.4.2.1	<a href="#">CREATION_CLASS_NAME</a>	39
5.4.2.2	<a href="#">LANENDPOINT_CAPTION</a>	39
5.4.2.3	<a href="#">LANENDPOINT_DESC</a>	39
5.4.2.4	<a href="#">MAXHOSTNAMELEN</a>	39
5.4.3	Enumeration Type Documentation	39
5.4.3.1	<a href="#">"@1</a>	40
5.4.3.2	<a href="#">"@2</a>	40
5.4.3.3	<a href="#">"@3</a>	40
5.4.3.4	<a href="#">"@4</a>	41

5.4.3.5	"@5	41
5.4.3.6	"@6	42
5.4.3.7	"@7	42
5.4.3.8	"@8	43
5.4.3.9	"@9	43
5.4.4	Function Documentation	44
5.4.4.1	changeLinkOPState	44
5.4.4.2	freeLANEndpoint	44
5.4.4.3	freeLANEndpointList	44
5.4.4.4	getALLLANEndpoint	44
5.4.4.5	getLANEndpoint	45
5.4.4.6	getLANEndpoints	45
5.4.4.7	nlInfoTOLanEP	45
5.4.4.8	nlListTOLanEPList	46
5.5	/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_Netlink.h File Reference	47
5.5.1	Detailed Description	48
5.5.2	Define Documentation	49
5.5.2.1	ARRAY_SIZE	49
5.5.2.2	FREE_SAFE	49
5.5.2.3	NL_SOCKET_RCV_BUFF_LEN	49
5.5.2.4	NL_SOCKET_SND_BUFF_LEN	49
5.5.2.5	NLMSG_TAIL	49
5.5.3	Function Documentation	50
5.5.3.1	nlAddAttrToMsg	50
5.5.3.2	nlAddAttrToMsg32	50
5.5.3.3	nlAddLinkToList	50
5.5.3.4	nlAddr_n2a	51
5.5.3.5	nlAddRouteToList	51
5.5.3.6	nlCloseSocket	51
5.5.3.7	nlCreateDefaultLinkInfo	51
5.5.3.8	nlCreateDefaultRtInfo	52
5.5.3.9	nlGenLinkFilter	52
5.5.3.10	nlGenRouteFilter	52
5.5.3.11	nlGetLinks	53
5.5.3.12	nlGetLinkTypePos	53
5.5.3.13	nlGetRoutes	53



5.5.3.14	nlModifyLink	53
5.5.3.15	nlModifyRoute	54
5.5.3.16	nlOpenSocket	54
5.5.3.17	nlResetLinkFilter	54
5.5.3.18	nlResetRouteFilter	54
5.5.4	Variable Documentation	54
5.5.4.1	address	54
5.5.4.2	broadcast	55
5.5.4.3	change	55
5.5.4.4	dstAddr	55
5.5.4.5	dstLen	55
5.5.4.6	family	55
5.5.4.7	flags	55
5.5.4.8	gw	55
5.5.4.9	ifname	55
5.5.4.10	index	55
5.5.4.11	inputIf	55
5.5.4.12	link	55
5.5.4.13	linkFlt	56
5.5.4.14	linkmode	56
5.5.4.15	linkType	56
5.5.4.16	linkTypeName	56
5.5.4.17	map	57
5.5.4.18	metrics	57
5.5.4.19	mtu	57
5.5.4.20	operstate	57
5.5.4.21	outputIf	57
5.5.4.22	prefSrc	57
5.5.4.23	priority	57
5.5.4.24	protocol	57
5.5.4.25	qdisc	57
5.5.4.26	rtFlt	57
5.5.4.27	scope	57
5.5.4.28	srcAddr	58
5.5.4.29	srcLen	58
5.5.4.30	stats	58

5.5.4.31	table	58
5.5.4.32	tos	58
5.5.4.33	txqlen	58
5.5.4.34	type	58
5.6	/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_NextHopIPRoute.h File Reference	59
5.6.1	Detailed Description	60
5.6.2	Define Documentation	60
5.6.2.1	INSTANCEID_FORMAT	60
5.6.2.2	INSTANCEID_FORMAT_PARSE	60
5.6.3	Enumeration Type Documentation	61
5.6.3.1	"@12	61
5.6.3.2	"@13	61
5.6.3.3	"@14	61
5.6.3.4	"@15	62
5.6.3.5	"@16	62
5.6.3.6	"@17	62
5.6.4	Function Documentation	63
5.6.4.1	addIPRoute	63
5.6.4.2	delIPRoute	63
5.6.4.3	freeNextHopIP	63
5.6.4.4	freeNextHopIPList	63
5.6.4.5	getAddrType	64
5.6.4.6	getAddrTypeStr	64
5.6.4.7	getAllIPRoutes	64
5.6.4.8	getIPRouteId	64
5.6.4.9	getIPRoutes	65
5.6.4.10	nhTONlInfo	65
5.6.4.11	nlInfoTONh	65
5.6.4.12	nlListTONhList	65
5.7	/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_Zebra.h File Reference	67
5.7.1	Detailed Description	67
5.8	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_CSHostedRouteProvider.c File Reference	68
5.8.1	Detailed Description	69
5.8.2	Function Documentation	70
5.8.2.1	CMAssociationMISub	70
5.8.2.2	CMInstanceMISub	70

5.8.2.3	OSBase_CSHostedRouteProviderAssociationCleanup . . . . .	70
5.8.2.4	OSBase_CSHostedRouteProviderAssociatorNames . . . . .	70
5.8.2.5	OSBase_CSHostedRouteProviderAssociators . . . . .	70
5.8.2.6	OSBase_CSHostedRouteProviderCleanup . . . . .	70
5.8.2.7	OSBase_CSHostedRouteProviderCreateInstance . . . . .	70
5.8.2.8	OSBase_CSHostedRouteProviderDeleteInstance . . . . .	70
5.8.2.9	OSBase_CSHostedRouteProviderEnumInstanceNames . . . . .	71
5.8.2.10	OSBase_CSHostedRouteProviderEnumInstances . . . . .	71
5.8.2.11	OSBase_CSHostedRouteProviderExecQuery . . . . .	71
5.8.2.12	OSBase_CSHostedRouteProviderGetInstance . . . . .	71
5.8.2.13	OSBase_CSHostedRouteProviderReferenceNames . . . . .	71
5.8.2.14	OSBase_CSHostedRouteProviderReferences . . . . .	71
5.8.2.15	OSBase_CSHostedRouteProviderSetInstance . . . . .	71
5.8.3	Variable Documentation . . . . .	71
5.8.3.1	_broker . . . . .	71
5.8.3.2	_ClassName . . . . .	72
5.8.3.3	_RefLeft . . . . .	72
5.8.3.4	_RefLeftClass . . . . .	72
5.8.3.5	_RefRight . . . . .	72
5.8.3.6	_RefRightClass . . . . .	72
5.9	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_LANEndpoint.c File Reference . . . . .	73
5.9.1	Detailed Description . . . . .	73
5.9.2	Function Documentation . . . . .	74
5.9.2.1	_makeInst_LANEndpoint . . . . .	74
5.9.2.2	_makePath_LANEndpoint . . . . .	74
5.10	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_LANEndpointProvider.c File Reference . . . . .	75
5.10.1	Detailed Description . . . . .	76
5.10.2	Function Documentation . . . . .	76
5.10.2.1	CMInstanceMISub . . . . .	76
5.10.2.2	CMMMethodMISub . . . . .	76
5.10.2.3	OSBase_LANEndpointProviderCleanup . . . . .	77
5.10.2.4	OSBase_LANEndpointProviderCreateInstance . . . . .	77
5.10.2.5	OSBase_LANEndpointProviderDeleteInstance . . . . .	77
5.10.2.6	OSBase_LANEndpointProviderEnumInstanceNames . . . . .	77
5.10.2.7	OSBase_LANEndpointProviderEnumInstances . . . . .	77
5.10.2.8	OSBase_LANEndpointProviderExecQuery . . . . .	77

5.10.2.9	OSBase_LANEndpointProviderGetInstance	77
5.10.2.10	OSBase_LANEndpointProviderInvokeMethod	77
5.10.2.11	OSBase_LANEndpointProviderMethodCleanup	78
5.10.2.12	OSBase_LANEndpointProviderSetInstance	78
5.10.3	Variable Documentation	78
5.10.3.1	_broker	78
5.11	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_NextHopIPRoute.c File Reference	79
5.11.1	Detailed Description	79
5.11.2	Function Documentation	80
5.11.2.1	_makeInst_NextHopIPRoute	80
5.11.2.2	_makePath_NextHopIPRoute	80
5.12	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_NextHopIPRouteProvider.c File Reference	81
5.12.1	Detailed Description	82
5.12.2	Function Documentation	82
5.12.2.1	CMInstanceMISub	82
5.12.2.2	CMMMethodMISub	82
5.12.2.3	getNextHopIPParams	83
5.12.2.4	OSBase_NextHopIPRouteProviderCleanup	83
5.12.2.5	OSBase_NextHopIPRouteProviderCreateInstance	83
5.12.2.6	OSBase_NextHopIPRouteProviderDeleteInstance	83
5.12.2.7	OSBase_NextHopIPRouteProviderEnumInstanceNames	83
5.12.2.8	OSBase_NextHopIPRouteProviderEnumInstances	83
5.12.2.9	OSBase_NextHopIPRouteProviderExecQuery	83
5.12.2.10	OSBase_NextHopIPRouteProviderGetInstance	84
5.12.2.11	OSBase_NextHopIPRouteProviderInvokeMethod	84
5.12.2.12	OSBase_NextHopIPRouteProviderMethodCleanup	84
5.12.2.13	OSBase_NextHopIPRouteProviderSetInstance	84
5.12.3	Variable Documentation	84
5.12.3.1	_broker	84
5.13	/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_RouteUsesEndpoint.c File Reference	85
5.13.1	Detailed Description	85
5.13.2	Function Documentation	86
5.13.2.1	_assoc_get_NextHopRoute_insts	86
5.13.2.2	_assoc_get_ProtocolEndpoint_insts	86
5.13.2.3	_assoc_RouteUsesEndpoint	87
5.13.2.4	_makeInst_RouteUsesEndpoint	87

5.13.2.5	<a href="#">_makePath_RouteUsesEndpoint</a>	88
5.14	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_RouteUsesEndpointProvider.c File Reference</a>	89
5.14.1	Detailed Description	90
5.14.2	Function Documentation	90
5.14.2.1	<a href="#">CMAssociationMISub</a>	91
5.14.2.2	<a href="#">CMInstanceMISub</a>	91
5.14.2.3	<a href="#">OSBase_RouteUsesEndpointProviderAssociationCleanup</a>	91
5.14.2.4	<a href="#">OSBase_RouteUsesEndpointProviderAssociatorNames</a>	91
5.14.2.5	<a href="#">OSBase_RouteUsesEndpointProviderAssociators</a>	91
5.14.2.6	<a href="#">OSBase_RouteUsesEndpointProviderCleanup</a>	91
5.14.2.7	<a href="#">OSBase_RouteUsesEndpointProviderCreateInstance</a>	91
5.14.2.8	<a href="#">OSBase_RouteUsesEndpointProviderDeleteInstance</a>	91
5.14.2.9	<a href="#">OSBase_RouteUsesEndpointProviderEnumInstanceNames</a>	92
5.14.2.10	<a href="#">OSBase_RouteUsesEndpointProviderEnumInstances</a>	92
5.14.2.11	<a href="#">OSBase_RouteUsesEndpointProviderExecQuery</a>	92
5.14.2.12	<a href="#">OSBase_RouteUsesEndpointProviderGetInstance</a>	92
5.14.2.13	<a href="#">OSBase_RouteUsesEndpointProviderReferenceNames</a>	92
5.14.2.14	<a href="#">OSBase_RouteUsesEndpointProviderReferences</a>	92
5.14.2.15	<a href="#">OSBase_RouteUsesEndpointProviderSetInstance</a>	92
5.14.3	Variable Documentation	92
5.14.3.1	<a href="#">_broker</a>	93
5.15	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_LANEndpoint.c File Reference</a>	94
5.15.1	Detailed Description	94
5.15.2	Function Documentation	95
5.15.2.1	<a href="#">changeLinkOPState</a>	95
5.15.2.2	<a href="#">datetime_str_interval_to_ms</a>	95
5.15.2.3	<a href="#">freeLANEndpoint</a>	95
5.15.2.4	<a href="#">freeLANEndpointList</a>	96
5.15.2.5	<a href="#">getALLLANEndpoints</a>	96
5.15.2.6	<a href="#">getLANEndpoint</a>	96
5.15.2.7	<a href="#">getLANEndpoints</a>	96
5.15.2.8	<a href="#">nlInfoTOLanEP</a>	97
5.15.2.9	<a href="#">nlListTOLanEPList</a>	97
5.16	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_Netlink.c File Reference</a>	98
5.16.1	Detailed Description	99
5.16.2	Function Documentation	99

5.16.2.1	<a href="#">nlAddAttrToMsg</a>	99
5.16.2.2	<a href="#">nlAddAttrToMsg32</a>	99
5.16.2.3	<a href="#">nlAddLinkToList</a>	100
5.16.2.4	<a href="#">nlAddr_n2a</a>	100
5.16.2.5	<a href="#">nlAddRouteToList</a>	100
5.16.2.6	<a href="#">nlCloseSocket</a>	101
5.16.2.7	<a href="#">nlCreateDefaultLinkInfo</a>	101
5.16.2.8	<a href="#">nlCreateDefaultRtInfo</a>	101
5.16.2.9	<a href="#">nlGenLinkFilter</a>	101
5.16.2.10	<a href="#">nlGenRouteFilter</a>	102
5.16.2.11	<a href="#">nlGetLinks</a>	102
5.16.2.12	<a href="#">nlGetLinkTypePos</a>	102
5.16.2.13	<a href="#">nlGetRoutes</a>	103
5.16.2.14	<a href="#">nlModifyLink</a>	103
5.16.2.15	<a href="#">nlModifyRoute</a>	103
5.16.2.16	<a href="#">nlOpenSocket</a>	103
5.16.2.17	<a href="#">nlResetLinkFilter</a>	104
5.16.2.18	<a href="#">nlResetRouteFilter</a>	104
5.16.3	<a href="#">Variable Documentation</a>	104
5.16.3.1	<a href="#">nlSH</a>	104
5.17	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_NextHopIPRoute.c File Reference</a>	105
5.17.1	<a href="#">Detailed Description</a>	105
5.17.2	<a href="#">Function Documentation</a>	106
5.17.2.1	<a href="#">addIPRoute</a>	106
5.17.2.2	<a href="#">delIPRoute</a>	106
5.17.2.3	<a href="#">freeNextHopIP</a>	106
5.17.2.4	<a href="#">freeNextHopIPList</a>	106
5.17.2.5	<a href="#">getAddrType</a>	107
5.17.2.6	<a href="#">getAddrTypeStr</a>	107
5.17.2.7	<a href="#">getAllIPRoutes</a>	107
5.17.2.8	<a href="#">getIPRouteId</a>	107
5.17.2.9	<a href="#">getIPRoutes</a>	108
5.17.2.10	<a href="#">nhTONlInfo</a>	108
5.17.2.11	<a href="#">nlInfoTONh</a>	108
5.17.2.12	<a href="#">nlListTONhList</a>	108
5.18	<a href="#">/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_Zebra.c File Reference</a>	110

---

5.18.1 Detailed Description . . . . .	110
---------------------------------------	-----





# Chapter 1

## Todo List

Global [datetime\\_str\\_interval\\_to\\_ms](#) move this function to the proper file.

Class [LANEndpoint](#) how can obtain installation date form a device? char installDate[64];  
How can obtain device last change. char timeOfLastStateChange[64];

File [OSBase\\_Zebra.c](#) WILL BE USED IN THE FUTURE TO OBTAIN INFO FROM ZEBRA/QUAGGA.

File [OSBase\\_Zebra.h](#) WILL BE USED IN THE FUTURE TO OBTAIN INFO FROM ZEBRA/QUAGGA.



# Chapter 2

## Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

LANEndpoint . . . . .	7
LANEndpointList . . . . .	12
nextHopIP . . . . .	13
nextHopIPList . . . . .	16
nlLinkInfo . . . . .	17
nlLinkInfoList . . . . .	20
nlRouteInfo . . . . .	21
nlRouteInfoList . . . . .	24
nlSockHandle . . . . .	25



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_LANEndpoint.h . . . . .	27
/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_NextHopIPRoute.h . . . . .	30
/mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase_RouteUsesEndpoint.h . . . . .	32
/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_LANEndpoint.h . . . . .	37
/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_Netlink.h . . . . .	47
/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_NextHopIPRoute.h . . . . .	59
/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_Zebra.h . . . . .	67
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_CSHostedRouteProvider.c . . . . .	68
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_LANEndpoint.c . . . . .	73
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_LANEndpointProvider.c . . . . .	75
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_NextHopIPRoute.c . . . . .	79
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_NextHopIPRouteProvider.c . . . . .	81
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_RouteUsesEndpoint.c . . . . .	85
/mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase_RouteUsesEndpointProvider.c . . . . .	89
/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_LANEndpoint.c . . . . .	94
/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_Netlink.c . . . . .	98
/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_NextHopIPRoute.c . . . . .	105
/mnt/storage/TESIS/sblim/cmpi-router/src/OSBase_Zebra.c . . . . .	110



# Chapter 4

## Data Structure Documentation

### 4.1 LANEndpoint Struct Reference

```
#include <OSBase_LANEndpoint.h>
```

#### Data Fields

- char [aliasAddresses](#) [32]
- unsigned short [availReqStates](#) [9]
- char [caption](#) [64]
- unsigned short [communicationStatus](#)
- char [creationClassName](#) [256]
- char [description](#) [256]
- unsigned short [detailedStatus](#)
- char [elementName](#) [64]
- unsigned short [enabledDefault](#)
- unsigned short [enabledState](#)
- char [groupAddresses](#) [32]
- unsigned short [healthState](#)
- char [instanceID](#) [64]
- char [lanID](#) [64]
- char [macAddress](#) [32]
- unsigned int [maxDataSize](#)
- char [name](#) [256]
- char [nameFormat](#) [256]
- unsigned short [operatingStatus](#)
- unsigned short [operationalStatus](#)
- char [otherEnabledState](#) [64]
- char [otherTypeDescription](#) [64]
- unsigned short [primaryStatus](#)
- unsigned short [protocolIFType](#)
- unsigned short [requestedState](#)

- char [statusDescriptions](#) [256]
- char [systemCreationClassName](#) [256]
- char [systemName](#) [256]
- unsigned short [transitioningToState](#)

### 4.1.1 Detailed Description

This structure stores the LANEndpoint's properties.

#### Todo

how can obtain installation date form a device? char installDate[64];  
How can obtain device last change. char timeOfLastStateChange[64];

Definition at line 300 of file OSBase\_LANEndpoint.h.

### 4.1.2 Field Documentation

#### 4.1.2.1 char aliasAddresses[32]

Other unicast addresses that may be used to communicate with the [LANEndpoint](#).

Definition at line 301 of file OSBase\_LANEndpoint.h.

#### 4.1.2.2 unsigned short availReqStates[9]

The possible values for the RequestedState parameter of the method RequestStateChange.

Definition at line 304 of file OSBase\_LANEndpoint.h.

#### 4.1.2.3 char caption[64]

Short textual description of the object.

Definition at line 307 of file OSBase\_LANEndpoint.h.

#### 4.1.2.4 unsigned short communicationStatus

The ability of the instrumentation to communicate with the underlying ManagedElement.

Definition at line 309 of file OSBase\_LANEndpoint.h.

#### 4.1.2.5 char creationClassName[256]

The name of the class or the subclass used in the creation of an instance.

Definition at line 312 of file OSBase\_LANEndpoint.h.



**4.1.2.6 char description[256]**

Textual description of the object.

Definition at line 315 of file OSBase\_LANEndpoint.h.

**4.1.2.7 unsigned short detailedStatus**

Compliments PrimaryStatus with additional status detail.

Definition at line 317 of file OSBase\_LANEndpoint.h.

**4.1.2.8 char elementName[64]**

A user-friendly name for the object.

Definition at line 319 of file OSBase\_LANEndpoint.h.

**4.1.2.9 unsigned short enabledDefault**

Administrator's default configuration for the Enabled State.

Definition at line 321 of file OSBase\_LANEndpoint.h.

**4.1.2.10 unsigned short enabledState**

Indicates the enabled and disabled states of an element.

Definition at line 324 of file OSBase\_LANEndpoint.h.

**4.1.2.11 char groupAddresses[32]**

Multicast addresses to which the [LANEndpoint](#) listens.

Definition at line 326 of file OSBase\_LANEndpoint.h.

**4.1.2.12 unsigned short healthState**

Current health of the element.

Definition at line 328 of file OSBase\_LANEndpoint.h.

**4.1.2.13 char instanceID[64]**

Opaquely and uniquely identify an instance of this class.

Definition at line 329 of file OSBase\_LANEndpoint.h.

**4.1.2.14 char lanID[64]**

Identifier for the LAN Segment to which the Endpoint is connected.

Definition at line 331 of file OSBase\_LANEndpoint.h.

**4.1.2.15 char macAddress[32]**

The principal unicast address used in communication with the [LANEndpoint](#).

Definition at line 333 of file OSBase\_LANEndpoint.h.

**4.1.2.16 unsigned int maxDataSize**

The largest information field that may be sent or received by the [LANEndpoint](#).

Definition at line 336 of file OSBase\_LANEndpoint.h.

**4.1.2.17 char name[256]**

Identifies this ProtocolEndpoint.

Definition at line 339 of file OSBase\_LANEndpoint.h.

**4.1.2.18 char nameFormat[256]**

The naming heuristic that is selected to ensure that the value of the Name property is unique.

Definition at line 340 of file OSBase\_LANEndpoint.h.

**4.1.2.19 unsigned short operatingStatus**

Current status value for the operational condition of the element.

Definition at line 343 of file OSBase\_LANEndpoint.h.

**4.1.2.20 unsigned short operationalStatus**

Current statuses of the element.

Definition at line 346 of file OSBase\_LANEndpoint.h.

**4.1.2.21 char otherEnabledState[64]**

A string that describes the enabled or disabled state of the element.

Definition at line 347 of file OSBase\_LANEndpoint.h.

**4.1.2.22 char otherTypeDescription[64]**

Type of ProtocolEndpoint.

Definition at line 350 of file OSBase\_LANEndpoint.h.

**4.1.2.23 unsigned short primaryStatus**

High level status value.

Definition at line 351 of file OSBase\_LANEndpoint.h.

**4.1.2.24 unsigned short protocolIFType**

IANA ifType MIB.

Definition at line 352 of file OSBase\_LANEndpoint.h.

**4.1.2.25 unsigned short requestedState**

The last requested or desired state for the element.

Definition at line 353 of file OSBase\_LANEndpoint.h.

**4.1.2.26 char statusDescriptions[256]**

Strings describing the various OperationalStatus array values.

Definition at line 355 of file OSBase\_LANEndpoint.h.

**4.1.2.27 char systemCreationClassName[256]**

The CreationClassName of the scoping System.

Definition at line 357 of file OSBase\_LANEndpoint.h.

**4.1.2.28 char systemName[256]**

The Name of the scoping System.

Definition at line 359 of file OSBase\_LANEndpoint.h.

**4.1.2.29 unsigned short transitioningToState**

The target state to which the instance is transitioning.

Definition at line 360 of file OSBase\_LANEndpoint.h.

The documentation for this struct was generated from the following file:

- /mnt/storage/TESIS/sblim/cmpi-router/include/[OSBase\\_LANEndpoint.h](#)

## 4.2 LANEndpointList Struct Reference

```
#include <OSBase_LANEndpoint.h>
```

### Data Fields

- struct [LANEndpoint](#) \* `sptr`
- struct [LANEndpointList](#) \* `next`

### 4.2.1 Detailed Description

This structure is used as LANEndpoints list.

Definition at line 367 of file `OSBase_LANEndpoint.h`.

### 4.2.2 Field Documentation

#### 4.2.2.1 struct LANEndpointList\* next [read]

Pointer to next position in the list.

Definition at line 370 of file `OSBase_LANEndpoint.h`.

#### 4.2.2.2 struct LANEndpoint\* sptr [read]

Pointer to current position in the list.

Definition at line 368 of file `OSBase_LANEndpoint.h`.

The documentation for this struct was generated from the following file:

- `/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase_LANEndpoint.h`

## 4.3 nextHopIP Struct Reference

```
#include <OSBase_NextHopIPRoute.h>
```

### Data Fields

- unsigned short [addressType](#)
- unsigned short [adminDistance](#)
- char [caption](#) [64]
- char [description](#) [256]
- char [dstAddress](#) [64]
- char [dstMask](#) [64]
- char [elementName](#) [64]
- char [instanceID](#) [64]
- unsigned short [isStatic](#)
- char [otherDerivation](#) [64]
- unsigned char [prefixLength](#)
- unsigned short [routeDerivation](#)
- char [routeGateway](#) [64]
- unsigned short [routeMetric](#)
- unsigned short [routeOutputIf](#)
- unsigned short [routeScope](#)
- unsigned short [routeTable](#)
- unsigned short [routeType](#)
- unsigned short [typeOfRoute](#)

### 4.3.1 Detailed Description

This structure stores the route's properties.

Definition at line 132 of file OSBase\_NextHopIPRoute.h.

### 4.3.2 Field Documentation

#### 4.3.2.1 unsigned short addressType

The format of the address properties.

Definition at line 133 of file OSBase\_NextHopIPRoute.h.

#### 4.3.2.2 unsigned short adminDistance

Administrative distance of this route.

Definition at line 135 of file OSBase\_NextHopIPRoute.h.

**4.3.2.3 char caption[64]**

Short textual description of the object.

Definition at line 137 of file OSBase\_NextHopIPRoute.h.

**4.3.2.4 char description[256]**

Textual description of the object.

Definition at line 139 of file OSBase\_NextHopIPRoute.h.

**4.3.2.5 char dstAddress[64]**

Destination address to be reached.

Definition at line 141 of file OSBase\_NextHopIPRoute.h.

**4.3.2.6 char dstMask[64]**

The mask for the destination address.

Definition at line 143 of file OSBase\_NextHopIPRoute.h.

**4.3.2.7 char elementName[64]**

User-friendly name for the object.

Definition at line 145 of file OSBase\_NextHopIPRoute.h.

**4.3.2.8 char instanceID[64]**

Opaquely and uniquely identify an instance of this class.

Definition at line 147 of file OSBase\_NextHopIPRoute.h.

**4.3.2.9 unsigned short isStatic**

TRUE indicates that this is a static route.

Definition at line 149 of file OSBase\_NextHopIPRoute.h.

**4.3.2.10 char otherDerivation[64]**

A string describing how the route was derived.

Definition at line 151 of file OSBase\_NextHopIPRoute.h.

**4.3.2.11 unsigned char prefixLength**

The prefix length for the IPv6 destination address.

Definition at line 153 of file OSBase\_NextHopIPRoute.h.

**4.3.2.12 unsigned short routeDerivation**

How the route was derived.

Definition at line 155 of file OSBase\_NextHopIPRoute.h.

**4.3.2.13 char routeGateway[64]**

The gateway of the route.

Definition at line 156 of file OSBase\_NextHopIPRoute.h.

**4.3.2.14 unsigned short routeMetric**

Numeric indication as to the preference of this route.

Definition at line 157 of file OSBase\_NextHopIPRoute.h.

**4.3.2.15 unsigned short routeOutputIf**

Output interface index.

Definition at line 159 of file OSBase\_NextHopIPRoute.h.

**4.3.2.16 unsigned short routeScope**

Sort of distance to the destination.

Definition at line 160 of file OSBase\_NextHopIPRoute.h.

**4.3.2.17 unsigned short routeTable**

Routing table id.

Definition at line 162 of file OSBase\_NextHopIPRoute.h.

**4.3.2.18 unsigned short routeType**

Type of route.

Definition at line 163 of file OSBase\_NextHopIPRoute.h.

**4.3.2.19 unsigned short typeOfRoute**

Administrator Defined Route, Computed Route or Actual Route.

Definition at line 164 of file OSBase\_NextHopIPRoute.h.

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_NextHopIPRoute.h](#)

## 4.4 nextHopIPList Struct Reference

```
#include <OSBase_NextHopIPRoute.h>
```

### Data Fields

- struct [nextHopIP](#) \* [sptr](#)
- struct [nextHopIPList](#) \* [next](#)

### 4.4.1 Detailed Description

This structure is used as NextHopIPRoutes list.

Definition at line 171 of file OSBase\_NextHopIPRoute.h.

### 4.4.2 Field Documentation

#### 4.4.2.1 struct nextHopIPList\* next [read]

Pointer to next position in the list.

Definition at line 174 of file OSBase\_NextHopIPRoute.h.

#### 4.4.2.2 struct nextHopIP\* sptr [read]

Pointer to current position in the list.

Definition at line 172 of file OSBase\_NextHopIPRoute.h.

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_NextHopIPRoute.h](#)



## 4.5 nlLinkInfo Struct Reference

```
#include <OSBase_Netlink.h>
```

### Data Fields

- unsigned char [family](#)
- unsigned short [type](#)
- int [index](#)
- unsigned int [flags](#)
- unsigned int [change](#)
- unsigned char [address](#) [32]
- int [addressLen](#)
- unsigned char [broadcast](#) [32]
- int [broadcastLen](#)
- char [ifname](#) [IFNAMSIZ]
- unsigned int [mtu](#)
- int [link](#)
- char [qdisc](#) [64]
- struct net\_device\_stats [stats](#)
- int [txqlen](#)
- struct ifmap [map](#)
- unsigned char [operstate](#)
- unsigned char [linkmode](#)

### 4.5.1 Detailed Description

Information of a single link.

Definition at line 226 of file OSBase\_Netlink.h.

### 4.5.2 Field Documentation

#### 4.5.2.1 unsigned char address[32]

Interface L2 address (IFLA\_ADDRESS).

Definition at line 235 of file OSBase\_Netlink.h.

#### 4.5.2.2 int addressLen

Used to store the length of the address.

Definition at line 237 of file OSBase\_Netlink.h.

#### **4.5.2.3 unsigned char broadcast[32]**

L2 broadcast address (IFLA\_BROADCAST).

Definition at line 239 of file OSBase\_Netlink.h.

#### **4.5.2.4 int broadcastLen**

Used to store the length of the broadcast.

Definition at line 241 of file OSBase\_Netlink.h.

#### **4.5.2.5 unsigned int change**

Reserved for future use.

Definition at line 234 of file OSBase\_Netlink.h.

#### **4.5.2.6 unsigned char family**

AF\_UNSPEC (from ifinfomsg).

Definition at line 227 of file OSBase\_Netlink.h.

#### **4.5.2.7 unsigned int flags**

Device flags (see netdevice(7)).

Definition at line 232 of file OSBase\_Netlink.h.

#### **4.5.2.8 char ifname[IFNAMSIZ]**

Device name (IFLA\_IFNAME).

Definition at line 243 of file OSBase\_Netlink.h.

#### **4.5.2.9 int index**

Unique interface index.

Definition at line 231 of file OSBase\_Netlink.h.

#### **4.5.2.10 int link**

Link type (IFLA\_LINK).

Definition at line 247 of file OSBase\_Netlink.h.

#### **4.5.2.11 unsigned char linkmode**

Link mode.

Definition at line 262 of file OSBase\_Netlink.h.

**4.5.2.12 struct ifmap map [read]**

Device mapping structure.

Definition at line 258 of file OSBase\_Netlink.h.

**4.5.2.13 unsigned int mtu**

MTU of the device (IFLA\_MTU).

Definition at line 245 of file OSBase\_Netlink.h.

**4.5.2.14 unsigned char operstate**

Operational state (rfc 2863).

Definition at line 260 of file OSBase\_Netlink.h.

**4.5.2.15 char qdisc[64]**

Queueing discipline (IFLA\_QDISC).

Definition at line 248 of file OSBase\_Netlink.h.

**4.5.2.16 struct net\_device\_stats stats [read]**

Interface statistics (IFLA\_STATS).

Definition at line 250 of file OSBase\_Netlink.h.

**4.5.2.17 int txqlen**

Tx queue length.

Definition at line 257 of file OSBase\_Netlink.h.

**4.5.2.18 unsigned short type**

Link type - ARPHRD\_\* (from ifinfomsg).

Definition at line 229 of file OSBase\_Netlink.h.

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_Netlink.h](#)

## 4.6 nlLinkInfoList Struct Reference

```
#include <OSBase_Netlink.h>
```

### Data Fields

- struct [nlLinkInfo](#) \* [sptr](#)
- struct [nlLinkInfoList](#) \* [next](#)

### 4.6.1 Detailed Description

Links container.

Definition at line 273 of file OSBase\_Netlink.h.

### 4.6.2 Field Documentation

#### 4.6.2.1 struct nlLinkInfoList\* next [read]

Pointer to next position in the list.

Definition at line 276 of file OSBase\_Netlink.h.

#### 4.6.2.2 struct nlLinkInfo\* sptr [read]

Pointer to current position in the list.

Definition at line 274 of file OSBase\_Netlink.h.

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_Netlink.h](#)

## 4.7 nlRouteInfo Struct Reference

```
#include <OSBase_Netlink.h>
```

### Data Fields

- int [family](#)
- int [type](#)
- int [protocol](#)
- int [scope](#)
- int [srcLen](#)
- int [dstLen](#)
- int [tos](#)
- char [dstAddr](#) [64]
- char [srcAddr](#) [64]
- int [inputIf](#)
- int [outputIf](#)
- char [gw](#) [64]
- int [priority](#)
- char [prefSrc](#) [64]
- int [metrics](#)
- int [table](#)

### 4.7.1 Detailed Description

Information of a single route.

Definition at line 116 of file OSBase\_Netlink.h.

### 4.7.2 Field Documentation

#### 4.7.2.1 char dstAddr[64]

Destination address to be reached (RTA\_DST).

Definition at line 124 of file OSBase\_Netlink.h.

#### 4.7.2.2 int dstLen

Destination Route mask prefix (from rtmsg).

Definition at line 122 of file OSBase\_Netlink.h.

#### 4.7.2.3 int family

IPv4=AF\_INET ; IPv6=AF\_INET6 (from rtmsg).

Definition at line 117 of file OSBase\_Netlink.h.

#### 4.7.2.4 char gw[64]

Gateway of the route (RTA\_GATEWAY).

Definition at line 128 of file OSBase\_Netlink.h.

#### 4.7.2.5 int inputIf

Input interface index (RTA\_IIF).

Definition at line 126 of file OSBase\_Netlink.h.

#### 4.7.2.6 int metrics

Route metrics (RTA\_METRICS).

Definition at line 131 of file OSBase\_Netlink.h.

#### 4.7.2.7 int outputIf

Output interface index (RTA\_OIF).

Definition at line 127 of file OSBase\_Netlink.h.

#### 4.7.2.8 char prefSrc[64]

Preferred source (RTA\_PREFSRC).

Definition at line 130 of file OSBase\_Netlink.h.

#### 4.7.2.9 int priority

Priority of the route (RTA\_PRIORITY).

Definition at line 129 of file OSBase\_Netlink.h.

#### 4.7.2.10 int protocol

Route origin (from rtmsg).

Definition at line 119 of file OSBase\_Netlink.h.

#### 4.7.2.11 int scope

Distance to the destination (from rtmsg).

Definition at line 120 of file OSBase\_Netlink.h.

**4.7.2.12 char srcAddr[64]**

Source address (RTA\_SRC).

Definition at line 125 of file OSBase\_Netlink.h.

**4.7.2.13 int srcLen**

Source Route mask prefix (from rtmsg).

Definition at line 121 of file OSBase\_Netlink.h.

**4.7.2.14 int table**

Route table (RTA\_TABLE).

Definition at line 132 of file OSBase\_Netlink.h.

**4.7.2.15 int tos**

Type of service (from rtmsg).

Definition at line 123 of file OSBase\_Netlink.h.

**4.7.2.16 int type**

Route type (from rtmsg).

Definition at line 118 of file OSBase\_Netlink.h.

The documentation for this struct was generated from the following file:

- /mnt/storage/TESIS/sblim/cmpi-router/include/[OSBase\\_Netlink.h](#)

## 4.8 nlRouteInfoList Struct Reference

```
#include <OSBase_Netlink.h>
```

### Data Fields

- struct [nlRouteInfo](#) \* [sptr](#)
- struct [nlRouteInfoList](#) \* [next](#)

### 4.8.1 Detailed Description

Routes container.

Definition at line 138 of file [OSBase\\_Netlink.h](#).

### 4.8.2 Field Documentation

#### 4.8.2.1 struct [nlRouteInfoList](#)\* [next](#) [read]

Pointer to next position in the list.

Definition at line 141 of file [OSBase\\_Netlink.h](#).

#### 4.8.2.2 struct [nlRouteInfo](#)\* [sptr](#) [read]

Pointer to current position in the list.

Definition at line 139 of file [OSBase\\_Netlink.h](#).

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_Netlink.h](#)



## 4.9 nlSockHandle Struct Reference

```
#include <OSBase_Netlink.h>
```

### Data Fields

- int [fd](#)
- struct sockaddr\_nl [local](#)

### 4.9.1 Detailed Description

Netlink socket handler.

Definition at line 67 of file OSBase\_Netlink.h.

### 4.9.2 Field Documentation

#### 4.9.2.1 int fd

File descriptor for the new socket.

Definition at line 68 of file OSBase\_Netlink.h.

#### 4.9.2.2 struct sockaddr\_nl local [read]

Netlink client in user-space.

Definition at line 69 of file OSBase\_Netlink.h.

The documentation for this struct was generated from the following file:

- [/mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\\_Netlink.h](#)



# Chapter 5

## File Documentation

### 5.1 /mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase\_LANEndpoint.h File Reference

```
#include "cmpidt.h"
#include "OSBase_LANEndpoint.h"
```

#### Functions

- CMPIObjectPath \* [\\_makePath\\_LANEndpoint](#) (const CMPIBroker \*[\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const struct [LANEndpoint](#) \*pLANEP, CMPIStatus \*rc)
- CMPIInstance \* [\\_makeInst\\_LANEndpoint](#) (const CMPIBroker \*[\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const char \*\*properties, const struct [LANEndpoint](#) \*pLANEP, CMPIStatus \*rc)

#### Variables

- static char \* [\\_ClassName](#) = "Linux\_LANEndpoint"

#### 5.1.1 Detailed Description

[cmpiOSBase\\_LANEndpoint.h](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

**Contributors:**

**Description:**

This file defines the interfaces for the factory implementation of the CIM class Linux\_LANEndpoint.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_LANEndpoint.h](#).

## 5.1.2 Function Documentation

**5.1.2.1** `CMPIInstance* _makeInst_LANEndpoint (const CMPIBroker * _broker, const CMPIContext * ctx, const CMPIObjectPath * cop, const char ** properties, const struct LANEndpoint * pLANEP, CMPIStatus * rc)`

Method to create a CMPIInstance of this class.

**Parameters:**

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*properties* [in]  
*pLANEP* [in] [LANEndpoint](#) instance.  
*rc* [in] cim status.

**Returns:**

created [LANEndpoint](#) instance.

Definition at line 85 of file [cmpiOSBase\\_LANEndpoint.c](#).

**5.1.2.2** `CMPIObjectPath* _makePath_LANEndpoint (const CMPIBroker * _broker, const CMPIContext * ctx, const CMPIObjectPath * cop, const struct LANEndpoint * pLANEP, CMPIStatus * rc)`

Method to create a CMPIObjectPath of this class.

**Parameters:**

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*pLANEP* [in] [LANEndpoint](#) instance.  
*rc* [in] cim status.

**Returns:**

created [LANEndpoint](#) ObjectPath.

Definition at line 41 of file cmpiOSBase\_LANEndpoint.c.

### **5.1.3 Variable Documentation**

#### **5.1.3.1 `char* _ClassName = "Linux_LANEndpoint" [static]`**

Provider ClassName.

Definition at line 33 of file cmpiOSBase\_LANEndpoint.h.

## 5.2 /mnt/storage/TESIS/sblim/cmpi-router/include/cmpiOSBase\_NextHopIPRoute.h File Reference

```
#include "cmpidt.h"
#include "OSBase_NextHopIPRoute.h"
```

### Functions

- CMPIObjectPath \* [\\_makePath\\_NextHopIPRoute](#) (const CMPIBroker \* [\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const struct [nextHopIP](#) \*pNHop, CMPIStatus \*rc)
- CMPIInstance \* [\\_makeInst\\_NextHopIPRoute](#) (const CMPIBroker \* [\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const char \*\*properties, const struct [nextHopIP](#) \*pNHop, CMPIStatus \*rc)

### Variables

- static char \* [\\_ClassName](#) = "Linux\_NextHopIPRoute"

### 5.2.1 Detailed Description

[cmpiOSBase\\_NextHopIPRoute.h](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This file defines the interfaces for the factory implementation of the CIM class Linux\_NextHopIPRoute.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_NextHopIPRoute.h](#).

### 5.2.2 Function Documentation

### 5.2.2.1 **CMPIInstance\* \_makeInst\_NextHopIPRoute (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const char \*\* *properties*, const struct nextHopIP \* *pNHop*, CMPIStatus \* *rc*)**

Method to create a CMPIInstance of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*properties* [in]  
*pNHop* [in] [nextHopIP](#) instance.  
*rc* [in] cim status.

#### Returns:

created NextHopIPRoute instance.

Definition at line 82 of file `cmpiOSBase_NextHopIPRoute.c`.

### 5.2.2.2 **CMPIObjectPath\* \_makePath\_NextHopIPRoute (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const struct nextHopIP \* *pNHop*, CMPIStatus \* *rc*)**

Method to create a CMPIObjectPath of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*pNHop* [in] [nextHopIP](#) instance.  
*rc* [in] cim status.

#### Returns:

created NextHopIPRoute ObjectPath.

Definition at line 42 of file `cmpiOSBase_NextHopIPRoute.c`.

## 5.2.3 Variable Documentation

### 5.2.3.1 **char\* \_ClassName = "Linux\_NextHopIPRoute" [static]**

Provider ClassName.

Definition at line 33 of file `cmpiOSBase_NextHopIPRoute.h`.

## 5.3 /mnt/storage/TESIS/sblim/mpi-router/include/mpiOSBase\_RouteUsesEndpoint.h File Reference

```
#include "cmptdt.h"
```

### Enumerations

- enum { [ATYPE\\_ASSOC](#), [ATYPE ASSO CN](#), [ATYPE\\_REFER](#), [ATYPE REFER N](#) }

### Functions

- CMPIObjectPath \* [\\_makePath\\_RouteUsesEndpoint](#) (const CMPIBroker \* [\\_broker](#), const CMPIObjectPath \*ops, const CMPIObjectPath \*opt, CMPIStatus \*rc)
- CMPIInstance \* [\\_makeInst\\_RouteUsesEndpoint](#) (const CMPIBroker \* [\\_broker](#), const CMPIObjectPath \*ops, const CMPIObjectPath \*opt, CMPIStatus \*rc)
- CMPIStatus \* [\\_assoc\\_get\\_NextHopRoute\\_insts](#) (const CMPIBroker \* [\\_broker](#), const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref, const CMPIInstance \*sourceInst, const char \*targetClass, int assocType, CMPIStatus \*rc)
- CMPIStatus \* [\\_assoc\\_get\\_ProtocolEndpoint\\_insts](#) (const CMPIBroker \* [\\_broker](#), const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref, const CMPIInstance \*sourceInst, const char \*targetClass, int assocType, CMPIStatus \*rc)
- CMPIStatus [\\_assoc\\_RouteUsesEndpoint](#) (const CMPIBroker \* [\\_broker](#), const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref, const char \*targetClass, int assocType, CMPIStatus \*rc)

### Variables

- static char \* [\\_ClassName](#) = "Linux\_RouteUsesEndpoint"
- static char \* [\\_RefLeft](#) = "Antecedent"
- static char \* [\\_RefRight](#) = "Dependent"
- static char \* [\\_RefLeftClass](#) = "CIM\_ProtocolEndpoint"
- static char \* [\\_RefRightClass](#) = "CIM\_NextHopRoute"
- static char \* [\\_RefLeftClasses](#) [] = { "Linux\_LANEndpoint" }
- static char \* [\\_RefRightClasses](#) [] = { "Linux\_NextHopIPRoute" }

#### 5.3.1 Detailed Description

[cmptOSBase\\_RouteUsesEndpoint.h](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>



**Author:**

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

**Contributors:**

**Description:**

This is the factory implementation for creating instances of CIM class Linux\_RouteUsesEndpoint.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_RouteUsesEndpoint.h](#).

## 5.3.2 Enumeration Type Documentation

### 5.3.2.1 anonymous enum

Association type.

**Enumerator:**

*ATYPE\_ASSOC*  
*ATYPE ASSO CN*  
*ATYPE\_REFER*  
*ATYPE REFERN*

Definition at line 32 of file cmpiOSBase\_RouteUsesEndpoint.h.

## 5.3.3 Function Documentation

**5.3.3.1** `CMPIStatus* _assoc_get_NextHopRoute_insts (const CMPIBroker * _broker, const CMPIContext * ctx, const CMPIResult * rslt, const CMPIObjectPath * ref, const CMPIInstance * sourceInst, const char * targetClass, int assocType, CMPIStatus * rc)`

Retrieve a list of instances from target class (CIM\_NextHopRoute subclass), associated to source class.

**Parameters:**

*\_broker* [in] CIM Object Manager.  
*ctx* [in] context object.  
*rslt* [in] result.  
*ref* [in] source objectPath.  
*sourceInst* [in] source class instance.

*targetClass* [in] target class name.

*assocType* [in] association type.

*rc* [in] cim status.

#### Returns:

cim status.

Definition at line 120 of file cmpiOSBase\_RouteUsesEndpoint.c.

#### 5.3.3.2 **CMPIStatus\* \_assoc\_get\_ProtocolEndpoint\_insts (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const CMPIInstance \* *sourceInst*, const char \* *targetClass*, int *assocType*, CMPIStatus \* *rc*)**

Retrieve a list of instances from target class (CIM\_ProtocolEndpoint subclass), associated to source class.

#### Parameters:

*\_broker* [in] CIM Object Manager.

*ctx* [in] context object.

*rslt* [in] result.

*ref* [in] source objectPath.

*sourceInst* [in] source class instance.

*targetClass* [in] target class name.

*assocType* [in] association type.

*rc* [in] cim status.

#### Returns:

cim status.

Definition at line 242 of file cmpiOSBase\_RouteUsesEndpoint.c.

#### 5.3.3.3 **CMPIStatus \_assoc\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *targetClass*, int *assocType*, CMPIStatus \* *rc*)**

Create CMPIInstances of association.

#### Parameters:

*\_broker* [in] CIM Object Manager.

*ctx* [in] context object.

*rslt* [in] result.

*ref* [in] source objectPath.

*targetClass* [in] target class name.

*assocType* [in] association type.

*rc* [in] cim status.

#### Returns:

cim status.

Definition at line 368 of file cmpiOSBase\_RouteUsesEndpoint.c.

### 5.3.3.4 CMPIInstance\* \_makeInst\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIObjectPath \* *ops*, const CMPIObjectPath \* *opt*, CMPIStatus \* *rc*)

Method to create a CMPIInstance of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ops* [in] Source objectPath.  
*opt* [in] Target objectPath.  
*rc* [in] cim status.

#### Returns:

created Instance.

Definition at line 79 of file cmpiOSBase\_RouteUsesEndpoint.c.

### 5.3.3.5 CMPIObjectPath\* \_makePath\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIObjectPath \* *ops*, const CMPIObjectPath \* *opt*, CMPIStatus \* *rc*)

Method to create a CMPIObjectPath of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ops* [in] Source objectPath.  
*opt* [in] Target objectPath.  
*rc* [in] cim status.

#### Returns:

created ObjectPath.

Definition at line 41 of file cmpiOSBase\_RouteUsesEndpoint.c.

## 5.3.4 Variable Documentation

### 5.3.4.1 char\* \_ClassName = "Linux\_RouteUsesEndpoint" [static]

Definition at line 39 of file cmpiOSBase\_RouteUsesEndpoint.h.

### 5.3.4.2 char\* \_RefLeft = "Antecedent" [static]

Definition at line 40 of file cmpiOSBase\_RouteUsesEndpoint.h.

### 5.3.4.3 char\* \_RefLeftClass = "CIM\_ProtocolEndpoint" [static]

Definition at line 42 of file cmpiOSBase\_RouteUsesEndpoint.h.

**5.3.4.4** `char* _RefLeftClasses[] = { "Linux_LANEndpoint" } [static]`

Definition at line 44 of file `cmpiOSBase_RouteUsesEndpoint.h`.

**5.3.4.5** `char* _RefRight = "Dependent" [static]`

Definition at line 41 of file `cmpiOSBase_RouteUsesEndpoint.h`.

**5.3.4.6** `char* _RefRightClass = "CIM_NextHopRoute" [static]`

Definition at line 43 of file `cmpiOSBase_RouteUsesEndpoint.h`.

**5.3.4.7** `char* _RefRightClasses[] = { "Linux_NextHopIPRoute" } [static]`

Definition at line 45 of file `cmpiOSBase_RouteUsesEndpoint.h`.

## 5.4 /mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\_LANEndpoint.h File Reference

```
#include "OSBase_Netlink.h"
```

### Data Structures

- struct [LANEndpoint](#)
- struct [LANEndpointList](#)

### Defines

- #define [MAXHOSTNAMELEN](#) 256
- #define [LANENDPOINT\\_CAPTION](#) "LAN EndPoint"
- #define [CREATION\\_CLASS\\_NAME](#) "Linux\_LANEndpoint"
- #define [LANENDPOINT\\_DESC](#)

### Enumerations

- enum {  
[LEP\\_ES\\_UNKNOWN](#), [LEP\\_ES\\_OTHER](#), [LEP\\_ES\\_ENABLED](#), [LEP\\_ES\\_DISABLED](#),  
[LEP\\_ES\\_SHUTTING\\_DOWN](#), [LEP\\_ES\\_NOT\\_APPLICABLE](#), [LEP\\_ES\\_ENABLED\\_BUT\\_OFFLINE](#),  
[LEP\\_ES\\_IN\\_TEST](#),  
[LEP\\_ES\\_DEFERRED](#), [LEP\\_ES QUIESCE](#), [LEP\\_ES\\_STARTING](#) }
- enum {  
[LEP\\_ED\\_ENABLED](#) = 2, [LEP\\_ED\\_DISABLED](#), [LEP\\_ED\\_NOT\\_APPLICABLE](#) = 5,  
[LEP\\_ED\\_ENABLED\\_BUT\\_OFFLINE](#),  
[LEP\\_ED\\_NO\\_DEFAULT](#), [LEP\\_ED QUIESCE](#) = 9 }
- enum {  
[LEP\\_RS\\_UNKNOWN](#), [LEP\\_RS\\_ENABLED](#) = 2, [LEP\\_RS\\_DISABLED](#),  
[LEP\\_RS\\_SHUT\\_DOWN](#),  
[LEP\\_RS\\_NO\\_CHANGE](#), [LEP\\_RS\\_OFFLINE](#), [LEP\\_RS\\_TEST](#), [LEP\\_RS\\_DEFERRED](#),  
[LEP\\_RS QUIESCE](#), [LEP\\_RS\\_REBOOT](#), [LEP\\_RS\\_RESET](#), [LEP\\_RS\\_NOT\\_APPLICABLE](#) }
- enum {  
[LEP\\_CS\\_UNKNOWN](#), [LEP\\_CS\\_NOT\\_AVAILABLE](#), [LEP\\_CS\\_COMMUNICATION\\_OK](#),  
[LEP\\_CS\\_LOST\\_COMMUNICATION](#),  
[LEP\\_CS\\_NO\\_CONTACT](#) }
- enum {  
[LEP\\_DS\\_NOT\\_AVAILABLE](#), [LEP\\_DS\\_NO\\_ADDITIONAL\\_INFORMATION](#),  
[LEP\\_DS\\_STRESSED](#), [LEP\\_DS\\_PREDICTIVE\\_FAILURE](#),  
[LEP\\_DS\\_NON\\_RECOVERABLE\\_ERROR](#), [LEP\\_DS\\_SUPPORTING\\_ENTITY\\_IN\\_ERROR](#) }

- enum {
  - LEP\_HS\_UNKNOWN = 0, LEP\_HS\_OK = 5, LEP\_HS\_DEGRADED\_WARNING = 10,
  - LEP\_HS\_MINOR\_FAILURE = 15,
  - LEP\_HS\_MAJOR\_FAILURE = 20, LEP\_HS\_CRITICAL\_FAILURE = 25,
  - LEP\_HS\_NON\_RECOVERABLE\_ERROR = 30 }
- enum {
  - LEP\_OS\_UNKNOWN, LEP\_OS\_NOT\_AVAILABLE, LEP\_OS\_SERVICING,
  - LEP\_OS\_STARTING,
  - LEP\_OS\_STOPPING, LEP\_OS\_STOPPED, LEP\_OS\_ABORTED, LEP\_OS\_DORMANT,
  - LEP\_OS\_COMPLETED, LEP\_OS\_MIGRATING, LEP\_OS\_EMIGRATING,
  - LEP\_OS\_IMMIGRATING,
  - LEP\_OS\_SNAPSHOTTING, LEP\_OS\_SHUTTING\_DOWN, LEP\_OS\_IN\_TEST,
  - LEP\_OS\_TRANSITIONING,
  - LEP\_OS\_IN\_SERVICE }
- enum {
  - LEP\_OPS\_UNKNOWN, LEP\_OPS\_OTHER, LEP\_OPS\_OK, LEP\_OPS\_DEGRADED,
  - LEP\_OPS\_STRESSED, LEP\_OPS\_PREDICTIVE\_FAILURE, LEP\_OPS\_ERROR,
  - LEP\_OPS\_NON\_RECOVERABLE\_ERROR,
  - LEP\_OPS\_STARTING, LEP\_OPS\_STOPPING, LEP\_OPS\_STOPPED, LEP\_OPS\_IN\_SERVICE,
  - LEP\_OPS\_NO\_CONTACT, LEP\_OPS\_LOST\_COMMUNICATION, LEP\_OPS\_ABORTED,
  - LEP\_OPS\_DORMANT,
  - LEP\_OPS\_SUPPORTING\_ENTITY\_IN\_ERROR, LEP\_OPS\_COMPLETED,
  - LEP\_OPS\_POWER\_MODE }
- enum { LEP\_PS\_UNKNOWN, LEP\_PS\_OK, LEP\_PS\_DEGRADED, LEP\_PS\_ERROR }

## Functions

- int [getLANEndpoints](#) (struct [LANEndpointList](#) \*\*lanEPList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [getALLLANEndpoint](#) (struct [LANEndpointList](#) \*\*lanEPList)
- int [getLANEndpoint](#) (struct [LANEndpointList](#) \*\*lanEPList, const char \*linkName)
- unsigned short [changeLinkOPState](#) (const char \*linkName, unsigned short enabledState, unsigned long timeoutPeriod)
- int [nlListTOlanEPList](#) (struct [nlLinkInfoList](#) \*\*nlLinkInfoList, struct [LANEndpointList](#) \*\*lanEPList)
- int [nlInfoTOlanEP](#) (struct [nlLinkInfo](#) \*nlLinkInfo, struct [LANEndpoint](#) \*lanEP)
- void [freeLANEndpointList](#) (struct [LANEndpointList](#) \*lptr)
- void [freeLANEndpoint](#) (struct [LANEndpoint](#) \*sptr)

### 5.4.1 Detailed Description

#### [OSBase\\_LANEndpoint.h](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

**Author:**

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

**Contributors:****Description:**

This file defines the interfaces for the resource access implementation of the CIM class `Linux_LANEndpoint`.

Definition in file [OSBase\\_LANEndpoint.h](#).

## 5.4.2 Define Documentation

### 5.4.2.1 `#define CREATION_CLASS_NAME "Linux_LANEndpoint"`

CIM - Creation class name.

Definition at line 48 of file `OSBase_LANEndpoint.h`.

### 5.4.2.2 `#define LANENDPOINT_CAPTION "LAN EndPoint"`

CIM - [LANEndpoint](#) caption.

Definition at line 43 of file `OSBase_LANEndpoint.h`.

### 5.4.2.3 `#define LANENDPOINT_DESC`

**Value:**

```
"A communication endpoint which, when " \
  "its associated interface device is connected to a LAN, " \
  "may send and receive data frames."
```

CIM - [LANEndpoint](#) description.

Definition at line 53 of file `OSBase_LANEndpoint.h`.

### 5.4.2.4 `#define MAXHOSTNAMELEN 256`

Maximum hostName length.

Definition at line 37 of file `OSBase_LANEndpoint.h`.

## 5.4.3 Enumeration Type Documentation

#### 5.4.3.1 anonymous enum

CIM - EnabledState.

Enumerator:

*LEP\_ES\_UNKNOWN*

*LEP\_ES\_OTHER*

*LEP\_ES\_ENABLED* Indicates that the element is or could be executing commands, will process any queued commands, and queues new requests.

*LEP\_ES\_DISABLED* Indicates that the element will not execute commands and will drop any new requests.

*LEP\_ES\_SHUTTING\_DOWN* Indicates that the element is in the process of going to a Disabled state.

*LEP\_ES\_NOT\_APPLICABLE* Indicates the element does not support being enabled or disabled.

*LEP\_ES\_ENABLED\_BUT\_OFFLINE* Indicates that the element might be completing commands, and will drop any new requests.

*LEP\_ES\_IN\_TEST* Indicates that the element is in a test state.

*LEP\_ES\_DEFERRED* Indicates that the element might be completing commands, but will queue any new requests.

*LEP\_ES\_QUIESCE* Indicates that the element is enabled but in a restricted mode.

*LEP\_ES\_STARTING* Indicates that the element is in the process of going to an Enabled state.

Definition at line 60 of file OSBase\_LANEndpoint.h.

#### 5.4.3.2 anonymous enum

CIM - EnabledDefault.

Enumerator:

*LEP\_ED\_ENABLED* Default.

*LEP\_ED\_DISABLED*

*LEP\_ED\_NOT\_APPLICABLE*

*LEP\_ED\_ENABLED\_BUT\_OFFLINE*

*LEP\_ED\_NO\_DEFAULT*

*LEP\_ED\_QUIESCE*

Definition at line 91 of file OSBase\_LANEndpoint.h.

#### 5.4.3.3 anonymous enum

CIM - RequestedState & TransitioningToState.

Enumerator:

*LEP\_RS\_UNKNOWN* Indicates the last requested state for the element is unknown.

*LEP\_RS\_ENABLED*

*LEP\_RS\_DISABLED*



***LEP\_RS\_SHUT\_DOWN***

***LEP\_RS\_NO\_CHANGE*** Deprecated.

***LEP\_RS\_OFFLINE*** Change to "Enabled but Offline" status.

***LEP\_RS\_TEST***

***LEP\_RS\_DEFERRED***

***LEP\_RS\_QUIESCE***

***LEP\_RS\_REBOOT*** Reboot refers to doing a "Shut Down" and then moving to an "Enabled" state.

***LEP\_RS\_RESET*** Reset indicates that the element is first "Disabled" and then "Enabled".

***LEP\_RS\_NOT\_APPLICABLE***

Definition at line 103 of file OSBase\_LANEndpoint.h.

#### 5.4.3.4 anonymous enum

CIM - CommunicationStatus.

**Enumerator:**

***LEP\_CS\_UNKNOWN*** Indicates the implementation is in general capable of returning this property, but is unable to do so at this time.

***LEP\_CS\_NOT\_AVAILABLE*** Indicates that the implementation is capable of returning a value for this property, but not ever for this particular piece of hardware/software.

***LEP\_CS\_COMMUNICATION\_OK*** Indicates communication is established with the element, but does not convey any quality of service.

***LEP\_CS\_LOST\_COMMUNICATION*** Indicates that the Managed Element is known to exist and has been contacted successfully in the past, but is currently unreachable.

***LEP\_CS\_NO\_CONTACT*** Indicates that the monitoring system has knowledge of this element, but has never been able to establish communications with it.

Definition at line 124 of file OSBase\_LANEndpoint.h.

#### 5.4.3.5 anonymous enum

CIM - DetailedStatus.

**Enumerator:**

***LEP\_DS\_NOT\_AVAILABLE*** Indicates that the implementation is capable of returning a value for this property, but not ever for this particular piece of hardware/software.

***LEP\_DS\_NO\_ADDITIONAL\_INFORMATION*** Indicates that the element is functioning normally as indicated by PrimaryStatus = "OK".

***LEP\_DS\_STRESSED*** Indicates that the element is functioning, but needs attention.

***LEP\_DS\_PREDICTIVE\_FAILURE*** Indicates that an element is functioning normally but a failure is predicted in the near future.

***LEP\_DS\_NON\_RECOVERABLE\_ERROR*** Indicates that this element is in an error condition that requires human intervention.

***LEP\_DS\_SUPPORTING\_ENTITY\_IN\_ERROR*** Indicates that this element might be "OK" but that another element, on which it is dependent, is in error.

Definition at line 147 of file OSBase\_LANEndpoint.h.

#### 5.4.3.6 anonymous enum

CIM - HealthState.

Enumerator:

**LEP\_HS\_UNKNOWN** The implementation cannot report on HealthState at this time.

**LEP\_HS\_OK** The element is fully functional and is operating within normal operational parameters and without error.

**LEP\_HS\_DEGRADED\_WARNING** The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities.

**LEP\_HS\_MINOR\_FAILURE** All functionality is available but some might be degraded.

**LEP\_HS\_MAJOR\_FAILURE** The element is failing. It is possible that some or all of the functionality of this component is degraded or not working.

**LEP\_HS\_CRITICAL\_FAILURE** The element is non-functional and recovery might not be possible.

**LEP\_HS\_NON\_RECOVERABLE\_ERROR** The element has completely failed, and recovery is not possible.

Definition at line 173 of file OSBase\_LANEndpoint.h.

#### 5.4.3.7 anonymous enum

CIM - OperatingStatus.

Enumerator:

**LEP\_OS\_UNKNOWN** Indicates the implementation is in general capable of returning this property, but is unable to do so at this time.

**LEP\_OS\_NOT\_AVAILABLE** Indicates that the implementation is capable of returning a value for this property, but not ever for this particular piece of hardware/software.

**LEP\_OS\_SERVICING** Describes an element being configured, maintained, cleaned, or otherwise administered.

**LEP\_OS\_STARTING** Describes an element being initialized.

**LEP\_OS\_STOPPING** Describes an element being brought to an orderly stop.

**LEP\_OS\_STOPPED** Clean and orderly stop.

**LEP\_OS\_ABORTED** Abrupt stop.

**LEP\_OS\_DORMANT** Indicates that the element is inactive or quiesced.

**LEP\_OS\_COMPLETED** Indicates that the element has completed its operation.

**LEP\_OS\_MIGRATING** Element is being moved between host elements.

**LEP\_OS\_EMIGRATING** Element is being moved away from host element.

**LEP\_OS\_IMMIGRATING** Element is being moved to new host element.

**LEP\_OS\_SNAPSHOTTING**

**LEP\_OS\_SHUTTING\_DOWN** Describes an element being brought to an abrupt stop.

**LEP\_OS\_IN\_TEST** Element is performing test functions.

**LEP\_OS\_TRANSITIONING** Describes an element that is between states.

**LEP\_OS\_IN\_SERVICE** Describes an element that is in service and operational.

Definition at line 199 of file OSBase\_LANEndpoint.h.

#### 5.4.3.8 anonymous enum

CIM - OperationalStatus.

Enumerator:

***LEP\_OPS\_UNKNOWN***

***LEP\_OPS\_OTHER*** Other status.

***LEP\_OPS\_OK*** The element is fully functional and is operating within normal operational.

***LEP\_OPS\_DEGRADED*** The element is in working order and all functionality is provided. However, the element is not working to the best of its abilities.

***LEP\_OPS\_STRESSED*** The element is functioning, but needs attention.

***LEP\_OPS\_PREDICTIVE\_FAILURE*** Element is functioning nominally but predicting a failure in the near future.

***LEP\_OPS\_ERROR*** In error state.

***LEP\_OPS\_NON\_RECOVERABLE\_ERROR*** Indicates that this element is in an error condition that requires human intervention.

***LEP\_OPS\_STARTING*** Indicates that the element is in the process of going to an Enabled state.

***LEP\_OPS\_STOPPING*** Describes an element being brought to an orderly stop.

***LEP\_OPS\_STOPPED*** Clean and orderly stop.

***LEP\_OPS\_IN\_SERVICE*** Element being configured, maintained, cleaned, or otherwise administered.

***LEP\_OPS\_NO\_CONTACT*** The monitoring system has knowledge of this element, but has never been able to establish communications with it.

***LEP\_OPS\_LOST\_COMMUNICATION*** The ManagedSystem Element is known to exist and has been contacted successfully in the past, but is currently unreachable.

***LEP\_OPS\_ABORTED*** Abrupt stop.

***LEP\_OPS\_DORMANT*** The element is inactive or quiesced.

***LEP\_OPS\_SUPPORTING\_ENTITY\_IN\_ERROR*** This element might be "OK" but that another element, on which it is dependent, is in error.

***LEP\_OPS\_COMPLETED*** The element has completed its operation.

***LEP\_OPS\_POWER\_MODE*** The element has additional power model information.

Definition at line 235 of file OSBase\_LANEndpoint.h.

#### 5.4.3.9 anonymous enum

CIM - PrimaryStatus.

Enumerator:

***LEP\_PS\_UNKNOWN***

***LEP\_PS\_OK***

***LEP\_PS\_DEGRADED***

***LEP\_PS\_ERROR***

Definition at line 286 of file OSBase\_LANEndpoint.h.

## 5.4.4 Function Documentation

### 5.4.4.1 unsigned short changeLinkOPState (const char \* *linkName*, unsigned short *enabledState*, unsigned long *timeoutPeriod*)

Used to set device status.

#### Parameters:

*linkName* [in] link name.

*enabledState* [in] desired state for the device.

*timeoutPeriod* [in] maximum amount of time that the client expects the transition to the new state to take.

#### Returns:

0 = Completed with No Error 1 = Not Supported 2 = Unknown or Unspecified Error 3 = Cannot complete within Timeout Period 4 = Failed 5 = Invalid Parameter 6 = In Use 7..4095 = DMTF Reserved 4096 = Method Parameters Checked - Job Started 4097 = Invalid State Transition 4098 = Use of Timeout Parameter Not Supported 4099 = Busy 4100..32767 = Method Reserved 32768..65535 = Vendor Specific

Definition at line 157 of file OSBase\_LANEndpoint.c.

### 5.4.4.2 void freeLANEndpoint (struct LANEndpoint \* *sptr*)

This function is used to clean a [LANEndpoint](#) structure.

#### Parameters:

*sptr* [in] structure to be cleaned.

Definition at line 501 of file OSBase\_LANEndpoint.c.

### 5.4.4.3 void freeLANEndpointList (struct LANEndpointList \* *lptr*)

This function is used to clean a [LANEndpoint](#) list.

#### Parameters:

*lptr* [in] list to be cleaned.

Definition at line 478 of file OSBase\_LANEndpoint.c.

### 5.4.4.4 int getALLLANEndpoint (struct LANEndpointList \*\* *lanEPList*)

This functions is used to get whole LANEndpoints available in the system.

#### Parameters:

*lanEPList* [out] [LANEndpoint](#) struct to be filled.

**Returns:**

0=succesful | 1=fail

**5.4.4.5 int getLANEndpoint (struct LANEndpointList \*\* *lanEPList*, const char \* *linkName*)**

This functions is used to get a route base on specified InstanceID.

**Parameters:**

*lanEPList* [out] [LANEndpoint](#) struct to be filled.

*linkName* [in] link ifname.

**Returns:**

0=succesful | 1=fail

Definition at line 130 of file OSBase\_LANEndpoint.c.

**5.4.4.6 int getLANEndpoints (struct LANEndpointList \*\* *lanEPList*, const struct nlLinkInfo \* *nlLinkInfo*)**

This functions is used to get a list of LANEndpoints based on [nlLinkInfo](#) filter.

**Parameters:**

*lanEPList* [out] [LANEndpoint](#) struct to be filled.

*nlLinkInfo* [in] link filter info.

**Returns:**

0=succesful | 1=fail

Definition at line 58 of file OSBase\_LANEndpoint.c.

**5.4.4.7 int nlInfoTOLanEP (struct nlLinkInfo \* *nlLinkInfo*, struct LANEndpoint \* *lanEP*)**

Converts a [nlLinkInfo](#) structure to [LANEndpoint](#) structure.

**Parameters:**

*nlLinkInfo* [in] structure to be converted.

*lanEP* [out] converted structure.

**Returns:**

0=succesful | 1=fail

Definition at line 287 of file OSBase\_LANEndpoint.c.

**5.4.4.8** `int nlListTOLanEPList (struct nlLinkInfoList ** nlLinkInfoList, struct LANEndpointList ** lanEPList)`

Converts a [nlLinkInfoList](#) structure to [LANEndpointList](#) strucutre.

**Parameters:**

*nlLinkInfoList* [in] list to be converted.

*lanEPList* [out] converted list.

**Returns:**

0=succesful | 1=fail

Definition at line 244 of file OSBase\_LANEndpoint.c.

## 5.5 /mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\_Netlink.h File Reference

```
#include <sys/socket.h>
#include <linux/netlink.h>
#include <linux/netdevice.h>
#include <linux/if_arp.h>
```

### Data Structures

- struct [nlSockHandle](#)
- struct [nlRouteInfo](#)
- struct [nlRouteInfoList](#)
- struct [nlLinkInfo](#)
- struct [nlLinkInfoList](#)

### Defines

- #define [NL\\_SOCKET\\_SND\\_BUFF\\_LEN](#) 32768
- #define [NL\\_SOCKET\\_RCV\\_BUFF\\_LEN](#) 32768
- #define [ARRAY\\_SIZE](#)(arr) (sizeof(arr) / sizeof((arr)[0]))
- #define [FREE\\_SAFE](#)(ptr) { if(ptr != NULL) free(ptr); ptr = NULL; }
- #define [NLMSG\\_TAIL](#)(nmsg) (((struct rtattr\*) (((void\*) (nmsg)) + NLMSG\_ALIGN((nmsg)->nmsg\_len))))

### Functions

- int [nlOpenSocket](#) ()
- int [nlCloseSocket](#) ()
- int [nlAddAttrToMsg](#) (struct nlmsghdr \*h, int maxlen, int attrType, const void \*attrData, int attrByte-len)
- int [nlAddAttrToMsg32](#) (struct nlmsghdr \*h, int maxlen, int attrType, unsigned int attrData)
- int [nlGetRoutes](#) (struct [nlRouteInfoList](#) \*\*nlRtInfoList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlModifyRoute](#) (struct [nlRouteInfo](#) \*nlRtInfo, int hType, unsigned int hFlags)
- int [nlAddRouteToList](#) (const struct nlmsghdr \*rcvH, struct [nlRouteInfoList](#) \*\*nlRtInfoList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlGenRouteFilter](#) (struct [nlRouteInfo](#) \*nlRtInfo)
- void [nlResetRouteFilter](#) ()
- int [nlCreateDefaultRtInfo](#) (struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlGetLinks](#) (struct [nlLinkInfoList](#) \*\*nlLinkInfoList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [nlModifyLink](#) (struct [nlLinkInfo](#) \*nlLinkInfo, int hType, unsigned int hFlags)
- int [nlAddLinkToList](#) (const struct nlmsghdr \*rcvH, struct [nlLinkInfoList](#) \*\*nlLinkInfoList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [nlGenLinkFilter](#) (const struct [nlLinkInfo](#) \*nlLinkInfo)

- void [nlResetLinkFilter](#) ()
- int [nlCreateDefaultLinkInfo](#) (struct [nlLinkInfo](#) \*[nlLinkInfo](#))
- void [nlAddr\\_n2a](#) (const unsigned char \*addr, int alen, int [type](#), char \*buf, int blen)
- unsigned short [nlGetLinkTypePos](#) (unsigned short devType)

## Variables

- struct {
  - unsigned int [family](#): 1
  - unsigned int [type](#): 1
  - unsigned int [protocol](#): 1
  - unsigned int [scope](#): 1
  - unsigned int [srcLen](#): 1
  - unsigned int [dstLen](#): 1
  - unsigned int [tos](#): 1
  - unsigned int [dstAddr](#): 1
  - unsigned int [srcAddr](#): 1
  - unsigned int [inputIf](#): 1
  - unsigned int [outputIf](#): 1
  - unsigned int [gw](#): 1
  - unsigned int [priority](#): 1
  - unsigned int [prefSrc](#): 1
  - unsigned int [metrics](#): 1
  - unsigned int [table](#): 1
 } [rtFlt](#)
- struct {
  - unsigned int [family](#): 1
  - unsigned int [type](#): 1
  - unsigned int [index](#): 1
  - unsigned int [flags](#): 1
  - unsigned int [change](#): 1
  - unsigned int [address](#): 1
  - unsigned int [broadcast](#): 1
  - unsigned int [ifname](#): 1
  - unsigned int [mtu](#): 1
  - unsigned int [link](#): 1
  - unsigned int [qdisc](#): 1
  - unsigned int [stats](#): 1
  - unsigned int [txqlen](#): 1
  - unsigned int [map](#): 1
  - unsigned int [operstate](#): 1
  - unsigned int [linkmode](#): 1
 } [linkFlt](#)
- static const unsigned short [linkType](#) []
- static const char \* [linkTypeName](#) []

### 5.5.1 Detailed Description

OSBase\_NetLink.h



THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

**Author:**

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

**Contributors:**

**Description:**

Netlink support lib.

Definition in file [OSBase\\_Netlink.h](#).

## 5.5.2 Define Documentation

### 5.5.2.1 `#define ARRAY_SIZE(arr) (sizeof(arr) / sizeof((arr)[0]))`

Get the size of an array.

Definition at line 51 of file [OSBase\\_Netlink.h](#).

### 5.5.2.2 `#define FREE_SAFE(ptr) { if(ptr != NULL) free(ptr); ptr = NULL; }`

Used to safe clean pointers.

Definition at line 56 of file [OSBase\\_Netlink.h](#).

### 5.5.2.3 `#define NL_SOCKET_RCV_BUFF_LEN 32768`

Netlink socket receive buffer length.

Definition at line 46 of file [OSBase\\_Netlink.h](#).

### 5.5.2.4 `#define NL_SOCKET_SND_BUFF_LEN 32768`

Netlink socket send buffer length.

Definition at line 41 of file [OSBase\\_Netlink.h](#).

### 5.5.2.5 `#define NLMSG_TAIL(nmsg) ((struct rtattr*) (((void*) (nmsg)) + NLMSG_ALIGN((nmsg)->nlmsg_len)))`

Netlink attributes.

Definition at line 61 of file [OSBase\\_Netlink.h](#).

### 5.5.3 Function Documentation

#### 5.5.3.1 `int nlAddAttrToMsg (struct nlmsghdr * h, int maxlen, int attrType, const void * attrData, int attrBytelen)`

This function is used to add an attribute to a netlink message.

**Parameters:**

*h* [out] message.  
*maxlen* [in] message length.  
*attrType* [in] attribute type.  
*attrData* [in] attribute data.  
*attrBytelen* [in] attribute length.

**Returns:**

0=succesful | 1=fail

Definition at line 147 of file OSBase\_Netlink.c.

#### 5.5.3.2 `int nlAddAttrToMsg32 (struct nlmsghdr * h, int maxlen, int attrType, unsigned int attrData)`

This function is used to add an attribute to a netlink message.

**Parameters:**

*h* [out] message.  
*maxlen* [in] message length.  
*attrType* [in] attribute type.  
*attrData* [in] attribute data.

**Returns:**

0=succesful | 1=fail

Definition at line 175 of file OSBase\_Netlink.c.

#### 5.5.3.3 `int nlAddLinkToList (const struct nlmsghdr * rcvH, struct nlLinkInfoList ** nlLinkInfoList, const struct nlLinkInfo * nlLinkInfo)`

This function is used to add a link to specified links list.

**Parameters:**

*rcvH* [in] link message.  
*nlLinkInfoList* [out] list where link will be added.  
*nlLinkInfo* [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1490 of file OSBase\_Netlink.c.

**5.5.3.4 void nlAddr\_n2a (const unsigned char \* *addr*, int *alen*, int *type*, char \* *buf*, int *blen*)**

Function to convert a L2 address from network representation to string representation (user friendly).

**Parameters:**

*addr* [in] L2 address to convert.  
*alen* [in] L2 address length.  
*type* [in] link type.  
*buf* [out] converted address.  
*blen* [in] converted address maximum length.

Definition at line 1892 of file OSBase\_Netlink.c.

**5.5.3.5 int nlAddRouteToList (const struct nlmsghdr \* *rcvH*, struct nlRouteInfoList \*\* *nlRtInfoList*, struct nlRouteInfo \* *nlRtInfo*)**

This function is used to add a route to specified routes list.

**Parameters:**

*rcvH* [in] route message.  
*nlRtInfoList* [out] list where route will be added.  
*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 659 of file OSBase\_Netlink.c.

**5.5.3.6 int nlCloseSocket ()**

Used to close previously open Netlink socket.

**Returns:**

0=succesful | 1=fail

Definition at line 124 of file OSBase\_Netlink.c.

**5.5.3.7 int nlCreateDefaultLinkInfo (struct nlLinkInfo \* *nlLinkInfo*)**

Used to create a [nlLinkInfo](#) structure with default values.

**Parameters:**

*nlLinkInfo* [out] link structure.

**Returns:**

0=succesful | 1=fail

Definition at line 1855 of file OSBase\_Netlink.c.

**5.5.3.8 int nlCreateDefaultRtInfo (struct nlRouteInfo \* *nlRtInfo*)**

Used to create a *nlRouteInfo* structure with default values.

**Parameters:**

*nlRtInfo* [out] route structure.

**Returns:**

0=succesful | 1=fail

Definition at line 1069 of file OSBase\_Netlink.c.

**5.5.3.9 int nlGenLinkFilter (const struct nlLinkInfo \* *nlLinkInfo*)**

Used to generate a filter to be applied to a list of links.

**Parameters:**

*nlLinkInfo* [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1700 of file OSBase\_Netlink.c.

**5.5.3.10 int nlGenRouteFilter (struct nlRouteInfo \* *nlRtInfo*)**

Used to generate a filter to be applied to a list of routes.

**Parameters:**

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 851 of file OSBase\_Netlink.c.

**5.5.3.11 int nlGetLinks (struct nlLinkInfoList \*\* *nlLinkInfoList*, const struct nlLinkInfo \* *nlLinkInfo*)**

This function is used to get links using [nlLinkInfo](#) as filter.

**Parameters:**

[nlLinkInfoList](#) [out] links list (one or more entries).

[nlLinkInfo](#) [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1110 of file OSBase\_Netlink.c.

**5.5.3.12 unsigned short nlGetLinkTypePos (unsigned short *devType*)**

Lookup the position of the passed type into linkType array.

**Parameters:**

*devType* [in] type to find.

**Returns:**

position into linkType array.

Definition at line 1929 of file OSBase\_Netlink.c.

**5.5.3.13 int nlGetRoutes (struct nlRouteInfoList \*\* *nlRtInfoList*, struct nlRouteInfo \* *nlRtInfo*)**

This function is used to get routes using *nlRtInfo* as filter.

**Parameters:**

*nlRtInfoList* [out] routes list (one or more entries).

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 209 of file OSBase\_Netlink.c.

**5.5.3.14 int nlModifyLink (struct nlLinkInfo \* *nlLinkInfo*, int *hType*, unsigned int *hFlags*)**

This function is used to ADD/DELETE/MODIFY a link.

**Parameters:**

[nlLinkInfo](#) [in] link info.

*hType* [in] type of message.

*hFlags* [in] message flags.

**Returns:**

0=succesful | 1=fail

Definition at line 1279 of file OSBase\_Netlink.c.

**5.5.3.15 int nlModifyRoute (struct nlRouteInfo \* *nlRtInfo*, int *hType*, unsigned int *hFlags*)**

This function is used to ADD/DELETE/MODIFY a route.

**Parameters:**

*nlRtInfo* [in] route info.

*hType* [in] type of message.

*hFlags* [in] message flags.

**Returns:**

0=succesful | 1=fail

Definition at line 384 of file OSBase\_Netlink.c.

**5.5.3.16 int nlOpenSocket ()**

Used to open a Netlink socket.

**Returns:**

0=succesful | 1=fail

Definition at line 48 of file OSBase\_Netlink.c.

**5.5.3.17 void nlResetLinkFilter ()**

Used to reset link filter handler.

Definition at line 1825 of file OSBase\_Netlink.c.

**5.5.3.18 void nlResetRouteFilter ()**

Used to reset route filter handler.

Definition at line 1042 of file OSBase\_Netlink.c.

## **5.5.4 Variable Documentation**

**5.5.4.1 unsigned int address**

Definition at line 290 of file OSBase\_Netlink.h.

**5.5.4.2 unsigned int broadcast**

Definition at line 291 of file OSBase\_Netlink.h.

**5.5.4.3 unsigned int change**

Definition at line 289 of file OSBase\_Netlink.h.

**5.5.4.4 unsigned int dstAddr**

Definition at line 157 of file OSBase\_Netlink.h.

**5.5.4.5 unsigned int dstLen**

Definition at line 155 of file OSBase\_Netlink.h.

**5.5.4.6 unsigned int family**

Definition at line 150 of file OSBase\_Netlink.h.

**5.5.4.7 unsigned int flags**

Definition at line 288 of file OSBase\_Netlink.h.

**5.5.4.8 unsigned int gw**

Definition at line 161 of file OSBase\_Netlink.h.

**5.5.4.9 unsigned int ifname**

Definition at line 292 of file OSBase\_Netlink.h.

**5.5.4.10 unsigned int index**

Definition at line 287 of file OSBase\_Netlink.h.

**5.5.4.11 unsigned int inputIf**

Definition at line 159 of file OSBase\_Netlink.h.

**5.5.4.12 unsigned int link**

Definition at line 294 of file OSBase\_Netlink.h.

#### 5.5.4.13 struct { ... } linkFlt

Struct used as link filter. If a field is deactivated (= 0), it will be skipped.

#### 5.5.4.14 unsigned int linkmode

Definition at line 306 of file OSBase\_Netlink.h.

#### 5.5.4.15 const unsigned short linkType[] [static]

Initial value:

```
{
    ARPHRD_NETROM, ARPHRD_ETHER, ARPHRD_EETHER, ARPHRD_AX25,
    ARPHRD_PRONET, ARPHRD_CHAOS, ARPHRD_IEEE802, ARPHRD_ARCNET,
    ARPHRD_APPLETLK, ARPHRD_DLCI, ARPHRD_ATM, ARPHRD_METRICOM,
    ARPHRD_IEEE1394, ARPHRD_EUI64, ARPHRD_INFINIBAND, ARPHRD_SLIP,
    ARPHRD_CSLIP, ARPHRD_SLIP6, ARPHRD_CSLIP6, ARPHRD_RSRVD,
    ARPHRD_ADAPT, ARPHRD_ROSE, ARPHRD_X25, ARPHRD_HWX25, ARPHRD_CAN,
    ARPHRD_PPP, ARPHRD_CISCO, ARPHRD_HDLC, ARPHRD_LAPB, ARPHRD_DDCMP,
    ARPHRD_RAWHDLC, ARPHRD_TUNNEL, ARPHRD_TUNNEL6, ARPHRD_FRAD,
    ARPHRD_SKIP, ARPHRD_LOOPBACK, ARPHRD_LOCALTLK, ARPHRD_FDDI,
    ARPHRD_BIF, ARPHRD_SIT, ARPHRD_IPDDP, ARPHRD_IPGRE,
    ARPHRD_PIMREG, ARPHRD_HIPPI, ARPHRD_ASH, ARPHRD_ECONET,
    ARPHRD_IRDA, ARPHRD_FCPP, ARPHRD_FCAL, ARPHRD_FCPL,
    ARPHRD_FCFABRIC, ARPHRD_IEEE802_TR, ARPHRD_IEEE80211,
    ARPHRD_IEEE80211_PRISM, ARPHRD_IEEE80211_RADIOTAP, ARPHRD_VOID,
    ARPHRD_NONE
}
```

The list of link types. (taken from kernel/net/core/dev.c)

Definition at line 318 of file OSBase\_Netlink.h.

#### 5.5.4.16 const char\* linkTypeName[] [static]

Initial value:

```
{
    "NETROM", "ETHER", "EETHER", "AX25",
    "PRONET", "CHAOS", "IEEE802", "ARCNET",
    "APPLETLK", "DLCI", "ATM", "METRICOM",
    "IEEE1394", "EUI64", "INFINIBAND", "SLIP",
    "CSLIP", "SLIP6", "CSLIP6", "RSRVD",
    "ADAPT", "ROSE", "X25", "HWX25", "CAN",
    "PPP", "CISCO", "HDLC", "LAPB", "DDCMP",
    "RAWHDLC", "TUNNEL", "TUNNEL6", "FRAD",
    "SKIP", "LOOPBACK", "LOCALTLK", "FDDI",
    "BIF", "SIT", "IPDDP", "IPGRE",
    "PIMREG", "HIPPI", "ASH", "ECONET",
    "IRDA", "FCPP", "FCAL", "FCPL",
    "FCFABRIC", "IEEE802_TR", "IEEE80211",
    "IEEE80211_PRISM", "IEEE80211_RADIOTAP",
    "VOID", "NONE"
}
```

The list of link types string representations. (taken from kernel/net/core/dev.c)

Definition at line 340 of file OSBase\_Netlink.h.



**5.5.4.17 unsigned int map**

Definition at line 303 of file OSBase\_Netlink.h.

**5.5.4.18 unsigned int metrics**

Definition at line 164 of file OSBase\_Netlink.h.

**5.5.4.19 unsigned int mtu**

Definition at line 293 of file OSBase\_Netlink.h.

**5.5.4.20 unsigned int operstate**

Definition at line 305 of file OSBase\_Netlink.h.

**5.5.4.21 unsigned int outputIf**

Definition at line 160 of file OSBase\_Netlink.h.

**5.5.4.22 unsigned int prefSrc**

Definition at line 163 of file OSBase\_Netlink.h.

**5.5.4.23 unsigned int priority**

Definition at line 162 of file OSBase\_Netlink.h.

**5.5.4.24 unsigned int protocol**

Definition at line 152 of file OSBase\_Netlink.h.

**5.5.4.25 unsigned int qdisc**

Definition at line 295 of file OSBase\_Netlink.h.

**5.5.4.26 struct { ... } rtFlt**

Struct used as route filter. If a field is deactivated (= 0), it will be skipped.

**5.5.4.27 unsigned int scope**

Definition at line 153 of file OSBase\_Netlink.h.

**5.5.4.28 unsigned int srcAddr**

Definition at line 158 of file OSBase\_Netlink.h.

**5.5.4.29 unsigned int srcLen**

Definition at line 154 of file OSBase\_Netlink.h.

**5.5.4.30 unsigned int stats**

Definition at line 296 of file OSBase\_Netlink.h.

**5.5.4.31 unsigned int table**

Definition at line 165 of file OSBase\_Netlink.h.

**5.5.4.32 unsigned int tos**

Definition at line 156 of file OSBase\_Netlink.h.

**5.5.4.33 unsigned int txqlen**

Definition at line 302 of file OSBase\_Netlink.h.

**5.5.4.34 unsigned int type**

Definition at line 151 of file OSBase\_Netlink.h.

## 5.6 /mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\_NextHopIPRoute.h File Reference

```
#include "OSBase_Netlink.h"
```

### Data Structures

- struct [nextHopIP](#)
- struct [nextHopIPList](#)

### Defines

- #define [INSTANCEID\\_FORMAT](#) "%s|%s|%d|%d|%d|%d|%s"
- #define [INSTANCEID\\_FORMAT\\_PARSE](#) "%[^|]|%[^|]|%d|%d|%d|%d|%[^|]"

### Enumerations

- enum { [NH\\_AT\\_UNKNOWN](#), [NH\\_AT\\_IPV4](#), [NH\\_AT\\_IPV6](#) }
- enum {  
[NH\\_RTD\\_UNKNOWN](#), [NH\\_RTD\\_OTHER](#), [NH\\_RTD\\_CONNECTED](#),  
[NH\\_RTD\\_USER\\_DEFINED](#),  
[NH\\_RTD\\_IGRP](#), [NH\\_RTD\\_EIGRP](#), [NH\\_RTD\\_RIP](#), [NH\\_RTD\\_HELLO](#),  
[NH\\_RTD\\_EGP](#), [NH\\_RTD\\_BGP](#), [NH\\_RTD\\_ISIS](#), [NH\\_RTD\\_OSPF](#) }
- enum {  
[NH\\_RTS\\_UNIVERSE](#) = 0, [NH\\_RTS\\_SITE](#) = 200, [NH\\_RTS\\_LINK](#) = 253, [NH\\_RTS\\_HOST](#) = 254,  
[NH\\_RTS\\_NOWHERE](#) = 255 }
- enum {  
[NH\\_RTT\\_UNSPEC](#) = 0, [NH\\_RTT\\_COMPAT](#) = 252, [NH\\_RTT\\_DEFAULT](#) = 253, [NH\\_RTT\\_MAIN](#)  
= 254,  
[NH\\_RTT\\_LOCAL](#) = 255 }
- enum {  
[NH\\_RTTY\\_UNSPEC](#), [NH\\_RTTY\\_UNICAST](#), [NH\\_RTTY\\_LOCAL](#), [NH\\_RTTY\\_BROADCAST](#),  
[NH\\_RTTY\\_ANYCAST](#), [NH\\_RTTY\\_MULTICAST](#), [NH\\_RTTY\\_BLACKHOLE](#),  
[NH\\_RTTY\\_UNREACHABLE](#),  
[NH\\_RTTY\\_PROHIBIT](#), [NH\\_RTTY\\_THROW](#), [NH\\_RTTY\\_NAT](#), [NH\\_RTTY\\_XRESOLVE](#) }
- enum { [NH\\_TOR\\_ADMINISTRATOR](#) = 2, [NH\\_TOR\\_COMPUTED](#), [NH\\_TOR\\_ACTUAL](#) }

### Functions

- int [getIPRoutes](#) (struct [nextHopIPList](#) \*\*nHopIPList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [getAllIPRoutes](#) (struct [nextHopIPList](#) \*\*nHopIPList)
- int [getIPRouteId](#) (struct [nextHopIPList](#) \*\*nHopIPList, char \*instanceId)
- int [addIPRoute](#) (struct [nextHopIP](#) \*nHopIP)

- int `delIPRoute` (struct `nextHopIP` \*nHopIP)
- int `nlListTOnhList` (struct `nlRouteInfoList` \*\*nlRtInfoList, struct `nextHopIPList` \*\*nHopIPList)
- int `nhTOnlInfo` (struct `nextHopIP` \*nHopIP, struct `nlRouteInfo` \*nlRtInfo)
- int `nlInfoTOnh` (struct `nlRouteInfo` \*nlRtInfo, struct `nextHopIP` \*nHopIP)
- int `getAddrType` (const char \*addr)
- int `getAddrTypeStr` (int addressType, char \*addrTypeStr)
- void `freeNextHopIPList` (struct `nextHopIPList` \*lptr)
- void `freeNextHopIP` (struct `nextHopIP` \*sptr)

### 5.6.1 Detailed Description

#### [OSBase\\_NextHopIPRoute.h](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This file defines the interfaces for the resource access implementation of the CIM class `Linux_NextHopIPRoute`.

Definition in file [OSBase\\_NextHopIPRoute.h](#).

### 5.6.2 Define Documentation

#### 5.6.2.1 `#define INSTANCEID_FORMAT "%s|%s|%d|%d|%d|%d|%s"`

Maximum `hostName` length. `InstanceID` format. It is supposed that it will be not changed.  
`hostName|dstAddr|dstLen|table|outputIf|scope|gw`

Definition at line 44 of file [OSBase\\_NextHopIPRoute.h](#).

#### 5.6.2.2 `#define INSTANCEID_FORMAT_PARSE "%[^]|%[^]|%d|%d|%d|%d|%^]"`

`InstanceID` format for parse. It is supposed that it will be not changed.

Definition at line 49 of file [OSBase\\_NextHopIPRoute.h](#).

## 5.6.3 Enumeration Type Documentation

### 5.6.3.1 anonymous enum

CIM - Describes the format of the address properties.

**Enumerator:**

*NH\_AT\_UNKNOWN* Unidentified.

*NH\_AT\_IPV4* IP version 4.

*NH\_AT\_IPV6* IP version 6.

Definition at line 54 of file OSBase\_NextHopIPRoute.h.

### 5.6.3.2 anonymous enum

CIM - How the route was derived.

**Enumerator:**

*NH\_RTD\_UNKNOWN*

*NH\_RTD\_OTHER*

*NH\_RTD\_CONNECTED*

*NH\_RTD\_USER\_DEFINED*

*NH\_RTD\_IGRP*

*NH\_RTD\_EIGRP*

*NH\_RTD\_RIP*

*NH\_RTD\_HELLO*

*NH\_RTD\_EGP*

*NH\_RTD\_BGP*

*NH\_RTD\_ISIS*

*NH\_RTD\_OSPF*

Definition at line 63 of file OSBase\_NextHopIPRoute.h.

### 5.6.3.3 anonymous enum

CIM - Route scope.

**Enumerator:**

*NH\_RTS\_UNIVERSE*

*NH\_RTS\_SITE*

*NH\_RTS\_LINK*

*NH\_RTS\_HOST*

*NH\_RTS\_NOWHERE*

Definition at line 81 of file OSBase\_NextHopIPRoute.h.

#### 5.6.3.4 anonymous enum

CIM - Routing table.

Enumerator:

*NH\_RTT\_UNSPEC*  
*NH\_RTT\_COMPAT*  
*NH\_RTT\_DEFAULT*  
*NH\_RTT\_MAIN* Main kernel table.  
*NH\_RTT\_LOCAL*

Definition at line 92 of file OSBase\_NextHopIPRoute.h.

#### 5.6.3.5 anonymous enum

CIM - Route Type.

Enumerator:

*NH\_RTTY\_UNSPEC*  
*NH\_RTTY\_UNICAST* Gateway or direct route.  
*NH\_RTTY\_LOCAL* Accept locally.  
*NH\_RTTY\_BROADCAST* Accept locally as broadcast, send as broadcast.  
*NH\_RTTY\_ANYCAST* Accept locally as broadcast, but send as unicast.  
*NH\_RTTY\_MULTICAST* Multicast route.  
*NH\_RTTY\_BLACKHOLE* Drop.  
*NH\_RTTY\_UNREACHABLE* Destination is unreachable.  
*NH\_RTTY\_PROHIBIT* Administratively prohibited.  
*NH\_RTTY\_THROW* Not in this table.  
*NH\_RTTY\_NAT* Translate this address.  
*NH\_RTTY\_XRESOLVE* Use external resolver.

Definition at line 103 of file OSBase\_NextHopIPRoute.h.

#### 5.6.3.6 anonymous enum

CIM - Type of route.

Enumerator:

*NH\_TOR\_ADMINISTRATOR*  
*NH\_TOR\_COMPUTED*  
*NH\_TOR\_ACTUAL*

Definition at line 123 of file OSBase\_NextHopIPRoute.h.

## 5.6.4 Function Documentation

### 5.6.4.1 int addIPRoute (struct nextHopIP \* *nHopIP*)

This function is used to add a new route to FIB.

**Parameters:**

*nHopIP* [in] route to be added.

**Returns:**

0=succesful | 1=fail

Definition at line 143 of file OSBase\_NextHopIPRoute.c.

### 5.6.4.2 int delIPRoute (struct nextHopIP \* *nHopIP*)

This function is used to delete a route from FIB.

**Parameters:**

*nHopIP* [in] route to be deleted.

**Returns:**

0=succesful | 1=fail

Definition at line 186 of file OSBase\_NextHopIPRoute.c.

### 5.6.4.3 void freeNextHopIP (struct nextHopIP \* *sptr*)

This function is used to clean a [nextHopIP](#) structure.

**Parameters:**

*sptr* [in] structure to be cleaned.

Definition at line 583 of file OSBase\_NextHopIPRoute.c.

### 5.6.4.4 void freeNextHopIPList (struct nextHopIPList \* *lptr*)

This function is used to clean a [nextHopIP](#) list.

**Parameters:**

*lptr* [in] list to be cleaned.

Definition at line 560 of file OSBase\_NextHopIPRoute.c.

#### 5.6.4.5 int getAddrType (const char \* *addr*)

Used to get CIM address type from specified route address.

**Parameters:**

*addr* [int] route address.

**Returns:**

CIM address type.

Definition at line 521 of file OSBase\_NextHopIPRoute.c.

#### 5.6.4.6 int getAddrTypeStr (int *addressType*, char \* *addrTypeStr*)

Used to get CIM address type in string format.

**Parameters:**

*addressType* [in] address type identifier.

*addrTypeStr* [out] address type string.

**Returns:**

0=succesful | 1=fail

Definition at line 540 of file OSBase\_NextHopIPRoute.c.

#### 5.6.4.7 int getAllIPRoutes (struct nextHopIPList \*\* *nHopIPList*)

This functions is used to get all ipv4 and ipv6 routes from fib.

**Parameters:**

*nHopIPList* [out] routes struct to be filled.

**Returns:**

0=succesful | 1=fail

Definition at line 77 of file OSBase\_NextHopIPRoute.c.

#### 5.6.4.8 int getIPRouteId (struct nextHopIPList \*\* *nHopIPList*, char \* *instanceId*)

This functions is used to get a route base on specified InstanceID.

**Parameters:**

*nHopIPList* [out] routes struct to be filled.

*instanceId* [in] route InstanceID.

**Returns:**

0=succesful | 1=fail

Definition at line 110 of file OSBase\_NextHopIPRoute.c.



**5.6.4.9 int getIPRoutes (struct nextHopIPList \*\* *nHopIPList*, struct nlRouteInfo \* *nlRtInfo*)**

This functions is used to get a list of routes based on *nlRtInfo* filter.

**Parameters:**

*nHopIPList* [out] routes struct to be filled.

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 31 of file OSBase\_NextHopIPRoute.c.

**5.6.4.10 int nhTONlInfo (struct nextHopIP \* *nHopIP*, struct nlRouteInfo \* *nlRtInfo*)**

Converts a [nextHopIP](#) structure to [nlRouteInfo](#) structure.

**Parameters:**

*nHopIP* [in] structure to be converted.

*nlRtInfo* [out] converted structure.

**Returns:**

0=succesful | 1=fail

Definition at line 273 of file OSBase\_NextHopIPRoute.c.

**5.6.4.11 int nlInfoTONh (struct nlRouteInfo \* *nlRtInfo*, struct nextHopIP \* *nHopIP*)**

Converts a [nlRouteInfo](#) structure to [nextHopIP](#) structure.

**Parameters:**

*nlRtInfo* [in] structure to be converted.

*nHopIP* [out] converted structure.

**Returns:**

0=succesful | 1=fail

Definition at line 381 of file OSBase\_NextHopIPRoute.c.

**5.6.4.12 int nlListTONhList (struct nlRouteInfoList \*\* *nlRtInfoList*, struct nextHopIPList \*\* *nHopIPList*)**

Converts a [nlRouteInfoList](#) structure to [nextHopIPList](#) strucutre.

**Parameters:**

*nlRtInfoList* [in] list to be converted.

*nHopIPList* [out] converted list.

**Returns:**

0=succesful | 1=fail

Definition at line 233 of file OSBase\_NextHopIPRoute.c.

## 5.7 /mnt/storage/TESIS/sblim/cmpi-router/include/OSBase\_Zebra.h File Reference

```
#include <sys/socket.h>
```

### 5.7.1 Detailed Description

#### OSBase\_Zebra.h

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

#### Todo

WILL BE USED IN THE FUTURE TO OBTAIN INFO FROM ZEBRA/QUAGGA.

Definition in file [OSBase\\_Zebra.h](#).

## 5.8 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_CSHostedRouteProvider.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpift.h"
#include "cmpimacs.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_Common.h"
```

### Functions

- **CMPIStatus OSBase\_CSHostedRouteProviderCleanup** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- **CMPIStatus OSBase\_CSHostedRouteProviderEnumInstanceNames** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref)
- **CMPIStatus OSBase\_CSHostedRouteProviderEnumInstances** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref, const char \*\*properties)
- **CMPIStatus OSBase\_CSHostedRouteProviderGetInstance** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*\*properties)
- **CMPIStatus OSBase\_CSHostedRouteProviderCreateInstance** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const CMPIInstance \*ci)
- **CMPIStatus OSBase\_CSHostedRouteProviderSetInstance** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const CMPIInstance \*ci, const char \*\*properties)
- **CMPIStatus OSBase\_CSHostedRouteProviderDeleteInstance** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop)
- **CMPIStatus OSBase\_CSHostedRouteProviderExecQuery** (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref, const char \*lang, const char \*query)
- **CMPIStatus OSBase\_CSHostedRouteProviderAssociationCleanup** (CMPIAssociationMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- **CMPIStatus OSBase\_CSHostedRouteProviderAssociators** (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*assocClass, const char \*resultClass, const char \*role, const char \*resultRole, const char \*\*propertyList)
- **CMPIStatus OSBase\_CSHostedRouteProviderAssociatorNames** (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*assocClass, const char \*resultClass, const char \*role, const char \*resultRole)
- **CMPIStatus OSBase\_CSHostedRouteProviderReferences** (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*assocClass, const char \*role, const char \*\*propertyList)
- **CMPIStatus OSBase\_CSHostedRouteProviderReferenceNames** (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*assocClass, const char \*role)

- [CMInstanceMIStub](#) (OSBase\_CSHostedRouteProvider, OSBase\_CSHostedRouteProvider, [\\_broker](#), CMNoHook)
- [CMAssociationMIStub](#) (OSBase\_CSHostedRouteProvider, OSBase\_CSHostedRouteProvider, [\\_broker](#), CMNoHook)

## Variables

- static const CMPIBroker \* [\\_broker](#)
- static char \* [\\_ClassName](#) = "Linux\_CSHostedRoute"
- static char \* [\\_RefLeft](#) = "Antecedent"
- static char \* [\\_RefRight](#) = "Dependent"
- static char \* [\\_RefLeftClass](#) = "Linux\_ComputerSystem"
- static char \* [\\_RefRightClass](#) = "CIM\_NextHopRoute"

### 5.8.1 Detailed Description

[cmpiOSBase\\_CSHostedRouteProvider.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

Linux\_CSHostedRoute association class provider implementation. The following CMPI instance methods are supported:

- Cleanup
- EnumerateInstanceNames
- EnumerateInstances
- GetInstance
- AssociationCleanup
- Associators
- AssociatorNames
- References
- ReferenceNames

Interface Type : Common Manageability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_CSHostedRouteProvider.c](#).

## 5.8.2 Function Documentation

**5.8.2.1** **CMAssociationMIStub** (OSBase\_CSHostedRouteProvider, OSBase\_CSHostedRouteProvider, \_broker, CMNoHook)

**5.8.2.2** **CMInstanceMIStub** (OSBase\_CSHostedRouteProvider, OSBase\_CSHostedRouteProvider, \_broker, CMNoHook)

**5.8.2.3** **CMPIStatus OSBase\_CSHostedRouteProviderAssociationCleanup** (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)

Definition at line 273 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.4** **CMPIStatus OSBase\_CSHostedRouteProviderAssociatorNames** (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *resultClass*, const char \* *role*, const char \* *resultRole*)

Definition at line 378 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.5** **CMPIStatus OSBase\_CSHostedRouteProviderAssociators** (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *resultClass*, const char \* *role*, const char \* *resultRole*, const char \*\* *propertyList*)

Definition at line 286 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.6** **CMPIStatus OSBase\_CSHostedRouteProviderCleanup** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)

Definition at line 65 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.7** **CMPIStatus OSBase\_CSHostedRouteProviderCreateInstance** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const CMPIInstance \* *ci*)

Definition at line 189 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.8** **CMPIStatus OSBase\_CSHostedRouteProviderDeleteInstance** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*)

Definition at line 229 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.2.9** **CMPIStatus OSBase\_CSHostedRouteProviderEnumInstanceNames** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*)

Definition at line 78 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.10** **CMPIStatus OSBase\_CSHostedRouteProviderEnumInstances** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \*\* *properties*)

Definition at line 115 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.11** **CMPIStatus OSBase\_CSHostedRouteProviderExecQuery** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *lang*, const char \* *query*)

Definition at line 247 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.12** **CMPIStatus OSBase\_CSHostedRouteProviderGetInstance** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \*\* *properties*)

Definition at line 152 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.13** **CMPIStatus OSBase\_CSHostedRouteProviderReferenceNames** (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *role*)

Definition at line 534 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.14** **CMPIStatus OSBase\_CSHostedRouteProviderReferences** (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *role*, const char \*\* *propertyList*)

Definition at line 469 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.2.15** **CMPIStatus OSBase\_CSHostedRouteProviderSetInstance** (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const CMPIInstance \* *ci*, const char \*\* *properties*)

Definition at line 208 of file mpiOSBase\_CSHostedRouteProvider.c.

## 5.8.3 Variable Documentation

**5.8.3.1** **const CMPIBroker\* \_broker** [static]

Definition at line 45 of file mpiOSBase\_CSHostedRouteProvider.c.

**5.8.3.2** `char* _ClassName = "Linux_CSHostedRoute" [static]`

Definition at line 51 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.3.3** `char* _RefLeft = "Antecedent" [static]`

Definition at line 52 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.3.4** `char* _RefLeftClass = "Linux_ComputerSystem" [static]`

Definition at line 54 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.3.5** `char* _RefRight = "Dependent" [static]`

Definition at line 53 of file `cmpiOSBase_CSHostedRouteProvider.c`.

**5.8.3.6** `char* _RefRightClass = "CIM_NextHopRoute" [static]`

Definition at line 55 of file `cmpiOSBase_CSHostedRouteProvider.c`.



## 5.9 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_LANEndpoint.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpimacs.h"
#include "cmpiOSBase_Common.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_LANEndpoint.h"
#include "OSBase_LANEndpoint.h"
```

### Functions

- `CMPIObjectPath * _makePath_LANEndpoint` (const `CMPIBroker *_broker`, const `CMPIContext *ctx`, const `CMPIObjectPath *cop`, const struct `LANEndpoint *pLANEP`, `CMPIStatus *rc`)
- `CMPIInstance * _makeInst_LANEndpoint` (const `CMPIBroker *_broker`, const `CMPIContext *ctx`, const `CMPIObjectPath *cop`, const char \*\*properties, const struct `LANEndpoint *pLANEP`, `CMPIStatus *rc`)

### 5.9.1 Detailed Description

`cmpiOSBase_LANEndpoint.c`

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This is the factory implementation for creating instances of CIM class `Linux_LANEndpoint`.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file `cmpiOSBase_LANEndpoint.c`.

## 5.9.2 Function Documentation

### 5.9.2.1 CMPIInstance\* \_makeInst\_LANEndpoint (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const char \*\* *properties*, const struct LANEndpoint \* *pLANEP*, CMPIStatus \* *rc*)

Method to create a CMPIInstance of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*properties* [in]  
*pLANEP* [in] [LANEndpoint](#) instance.  
*rc* [in] cim status.

#### Returns:

created [LANEndpoint](#) instance.

Definition at line 85 of file cmpiOSBase\_LANEndpoint.c.

### 5.9.2.2 CMPIObjectPath\* \_makePath\_LANEndpoint (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const struct LANEndpoint \* *pLANEP*, CMPIStatus \* *rc*)

Method to create a CMPIObjectPath of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*pLANEP* [in] [LANEndpoint](#) instance.  
*rc* [in] cim status.

#### Returns:

created [LANEndpoint](#) ObjectPath.

Definition at line 41 of file cmpiOSBase\_LANEndpoint.c.

## 5.10 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_LANEndpointProvider.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpift.h"
#include "cmpimacs.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_Common.h"
#include "cmpiOSBase_LANEndpoint.h"
```

### Functions

- **CMPIStatus** [OSBase\\_LANEndpointProviderCleanup](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- **CMPIStatus** [OSBase\\_LANEndpointProviderEnumInstanceNames](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref)
- **CMPIStatus** [OSBase\\_LANEndpointProviderEnumInstances](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref, const char \*\*properties)
- **CMPIStatus** [OSBase\\_LANEndpointProviderGetInstance](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const char \*\*properties)
- **CMPIStatus** [OSBase\\_LANEndpointProviderCreateInstance](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const CMPIInstance \*ci)
- **CMPIStatus** [OSBase\\_LANEndpointProviderSetInstance](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop, const CMPIInstance \*ci, const char \*\*properties)
- **CMPIStatus** [OSBase\\_LANEndpointProviderDeleteInstance](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*cop)
- **CMPIStatus** [OSBase\\_LANEndpointProviderExecQuery](#) (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref, const char \*lang, const char \*query)
- **CMPIStatus** [OSBase\\_LANEndpointProviderMethodCleanup](#) (CMPIMethodMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- **CMPIStatus** [OSBase\\_LANEndpointProviderInvokeMethod](#) (CMPIMethodMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIOObjectPath \*ref, const char \*methodName, const CMPIArgs \*in, CMPIArgs \*out)
- **CMInstanceMIStub** (OSBase\_LANEndpointProvider, OSBase\_LANEndpointProvider, [\\_broker](#), CMNoHook)
- **CMMethodMIStub** (OSBase\_LANEndpointProvider, OSBase\_LANEndpointProvider, [\\_broker](#), CMNoHook)

## Variables

- static const CMPIBroker \* [\\_broker](#)

### 5.10.1 Detailed Description

[cmpiOSBase\\_LANEndpointProvider.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

Linux\_LANEndpoint class provider implementation. The following CMPI instance methods are supported:

- Cleanup
- EnumerateInstanceNames
- EnumerateInstances
- GetInstance
- MethodCleanup
- InvokeMethod

Interface Type : Common Manageability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_LANEndpointProvider.c](#).

### 5.10.2 Function Documentation

**5.10.2.1 CMInstanceMISub (OSBase\_LANEndpointProvider, OSBase\_LANEndpointProvider, \_broker, CMNoHook)**

**5.10.2.2 CMMethodMISub (OSBase\_LANEndpointProvider, OSBase\_LANEndpointProvider, \_broker, CMNoHook)**

**5.10.2.3 CMPIStatus OSBase\_LANEndpointProviderCleanup (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)**

Definition at line 55 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.4 CMPIStatus OSBase\_LANEndpointProviderCreateInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const CMPIInstance \* *ci*)**

Definition at line 264 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.5 CMPIStatus OSBase\_LANEndpointProviderDeleteInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*)**

Definition at line 303 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.6 CMPIStatus OSBase\_LANEndpointProviderEnumInstanceNames (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*)**

Definition at line 68 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.7 CMPIStatus OSBase\_LANEndpointProviderEnumInstances (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \*\* *properties*)**

Definition at line 132 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.8 CMPIStatus OSBase\_LANEndpointProviderExecQuery (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *lang*, const char \* *query*)**

Definition at line 321 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.9 CMPIStatus OSBase\_LANEndpointProviderGetInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \*\* *properties*)**

Definition at line 198 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.10 CMPIStatus OSBase\_LANEndpointProviderInvokeMethod (CMPIMethodMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *methodName*, const CMPIArgs \* *in*, CMPIArgs \* *out*)**

Definition at line 360 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.11** **CMPIStatus OSBase\_LANEndpointProviderMethodCleanup** (CMPIMethodMI \* *mi*,  
const CMPIContext \* *ctx*, CMPIBoolean *terminating*)

Definition at line 347 of file cmpiOSBase\_LANEndpointProvider.c.

**5.10.2.12** **CMPIStatus OSBase\_LANEndpointProviderSetInstance** (CMPIInstanceMI \* *mi*, const  
CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const  
CMPIInstance \* *ci*, const char \*\* *properties*)

Definition at line 283 of file cmpiOSBase\_LANEndpointProvider.c.

### 5.10.3 Variable Documentation

**5.10.3.1** **const CMPIBroker\* \_broker** [**static**]

Definition at line 41 of file cmpiOSBase\_LANEndpointProvider.c.

## 5.11 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_NextHopIPRoute.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpimacs.h"
#include "cmpiOSBase_Common.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_NextHopIPRoute.h"
#include "OSBase_NextHopIPRoute.h"
```

### Functions

- CMPIObjectPath \* [\\_makePath\\_NextHopIPRoute](#) (const CMPIBroker \*[\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const struct [nextHopIP](#) \*pNHop, CMPIStatus \*rc)
- CMPIInstance \* [\\_makeInst\\_NextHopIPRoute](#) (const CMPIBroker \*[\\_broker](#), const CMPIContext \*ctx, const CMPIObjectPath \*cop, const char \*\*properties, const struct [nextHopIP](#) \*pNHop, CMPIStatus \*rc)

### 5.11.1 Detailed Description

[cmpiOSBase\\_NextHopIPRoute.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This is the factory implementation for creating instances of CIM class Linux\_NextHopIPRoute.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_NextHopIPRoute.c](#).

## 5.11.2 Function Documentation

### 5.11.2.1 CMPIInstance\* \_makeInst\_NextHopIPRoute (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const char \*\* *properties*, const struct nextHopIP \* *pNHop*, CMPIStatus \* *rc*)

Method to create a CMPIInstance of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*properties* [in]  
*pNHop* [in] [nextHopIP](#) instance.  
*rc* [in] cim status.

#### Returns:

created NextHopIPRoute instance.

Definition at line 82 of file cmpiOSBase\_NextHopIPRoute.c.

### 5.11.2.2 CMPIObjectPath\* \_makePath\_NextHopIPRoute (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIObjectPath \* *cop*, const struct nextHopIP \* *pNHop*, CMPIStatus \* *rc*)

Method to create a CMPIObjectPath of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ctx* [in] Context object.  
*cop* [in] Source objectPath.  
*pNHop* [in] [nextHopIP](#) instance.  
*rc* [in] cim status.

#### Returns:

created NextHopIPRoute ObjectPath.

Definition at line 42 of file cmpiOSBase\_NextHopIPRoute.c.



## 5.12 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_NextHopIPRouteProvider.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpift.h"
#include "cmpimacs.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_Common.h"
#include "cmpiOSBase_NextHopIPRoute.h"
```

### Functions

- **CMPIStatus** [getNextHopIPParams](#) (const **CMPIInstance** \*ci, struct [nextHopIP](#) \*nHopIP, struct [nextHopIP](#) \*nHopIPPrev)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderCleanup](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, **CMPIBoolean** terminating)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderEnumInstanceNames](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*ref)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderEnumInstances](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*ref, const char \*\*properties)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderGetInstance](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*cop, const char \*\*properties)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderCreateInstance](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*cop, const **CMPIInstance** \*ci)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderSetInstance](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*cop, const **CMPIInstance** \*ci, const char \*\*properties)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderDeleteInstance](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*cop)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderExecQuery](#) (**CMPIInstanceMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*ref, const char \*lang, const char \*query)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderMethodCleanup](#) (**CMPIMethodMI** \*mi, const **CMPIContext** \*ctx, **CMPIBoolean** terminating)
- **CMPIStatus** [OSBase\\_NextHopIPRouteProviderInvokeMethod](#) (**CMPIMethodMI** \*mi, const **CMPIContext** \*ctx, const **CMPIResult** \*rslt, const **CMPIOObjectPath** \*ref, const char \*methodName, const **CMPIArgs** \*in, **CMPIArgs** \*out)
- **CMInstanceMISub** ([OSBase\\_NextHopIPRouteProvider](#), [OSBase\\_NextHopIPRouteProvider](#), [\\_broker](#), **CMNoHook**)
- **CMMethodMISub** ([OSBase\\_NextHopIPRouteProvider](#), [OSBase\\_NextHopIPRouteProvider](#), [\\_broker](#), **CMNoHook**)

## Variables

- static const CMPIBroker \* [\\_broker](#)

### 5.12.1 Detailed Description

[cmpiOSBase\\_NextHopIPRouteProvider.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

Linux\_NextHopIPRoute class provider implementation. The following CMPI instance methods are supported:

- Cleanup
- EnumerateInstanceNames
- EnumerateInstances
- GetInstance
- CreateInstance
- SetInstance
- DeleteInstance

Interface Type : Common Manageability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_NextHopIPRouteProvider.c](#).

### 5.12.2 Function Documentation

**5.12.2.1 CMInstanceMISub (OSBase\_NextHopIPRouteProvider, OSBase\_NextHopIPRouteProvider, \_broker, CMNoHook)**

**5.12.2.2 CMMethodMISub (OSBase\_NextHopIPRouteProvider, OSBase\_NextHopIPRouteProvider, \_broker, CMNoHook)**

### 5.12.2.3 **CMPIStatus getNextHopIPParams (const CMPIInstance \* *ci*, struct nextHopIP \* *nHopIP*, struct nextHopIP \* *nHopIPPrev*)**

Get parameters from CMPIInstance.

#### Parameters:

*ci* [in] CMPIInstance.

*nHopIP* [out] structure to be filled.

*nHopIPPrev* [in] previous route info.

#### Returns:

0=succesful | 1=fail

Definition at line 550 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.4 **CMPIStatus OSBase\_NextHopIPRouteProviderCleanup (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)**

Definition at line 66 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.5 **CMPIStatus OSBase\_NextHopIPRouteProviderCreateInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const CMPIInstance \* *ci*)**

Definition at line 278 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.6 **CMPIStatus OSBase\_NextHopIPRouteProviderDeleteInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*)**

Definition at line 403 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.7 **CMPIStatus OSBase\_NextHopIPRouteProviderEnumInstanceNames (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*)**

Definition at line 79 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.8 **CMPIStatus OSBase\_NextHopIPRouteProviderEnumInstances (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \*\* *properties*)**

Definition at line 144 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.2.9 **CMPIStatus OSBase\_NextHopIPRouteProviderExecQuery (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *lang*, const char \* *query*)**

Definition at line 454 of file cmpiOSBase\_NextHopIPRouteProvider.c.

**5.12.2.10** **CMPIStatus OSBase\_NextHopIPRouteProviderGetInstance** (CMPIInstanceMI \* *mi*,  
const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*,  
const char \*\* *properties*)

Definition at line 212 of file cmpiOSBase\_NextHopIPRouteProvider.c.

**5.12.2.11** **CMPIStatus OSBase\_NextHopIPRouteProviderInvokeMethod** (CMPIMethodMI \* *mi*,  
const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const  
char \* *methodName*, const CMPIArgs \* *in*, CMPIArgs \* *out*)

Definition at line 493 of file cmpiOSBase\_NextHopIPRouteProvider.c.

**5.12.2.12** **CMPIStatus OSBase\_NextHopIPRouteProviderMethodCleanup** (CMPIMethodMI \* *mi*,  
const CMPIContext \* *ctx*, CMPIBoolean *terminating*)

Definition at line 480 of file cmpiOSBase\_NextHopIPRouteProvider.c.

**5.12.2.13** **CMPIStatus OSBase\_NextHopIPRouteProviderSetInstance** (CMPIInstanceMI \* *mi*,  
const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*,  
const CMPIInstance \* *ci*, const char \*\* *properties*)

Definition at line 324 of file cmpiOSBase\_NextHopIPRouteProvider.c.

### 5.12.3 Variable Documentation

**5.12.3.1** **const CMPIBroker\* \_broker** [**static**]

Definition at line 42 of file cmpiOSBase\_NextHopIPRouteProvider.c.

## 5.13 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_RouteUsesEndpoint.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpimacs.h"
#include "cmpiOSBase_Common.h"
#include "OSBase_Common.h"
#include "OSBase_Netlink.h"
#include "cmpiOSBase_RouteUsesEndpoint.h"
```

### Functions

- `CMPIObjectPath * _makePath_RouteUsesEndpoint` (const `CMPIBroker * _broker`, const `CMPIObjectPath *ops`, const `CMPIObjectPath *opt`, `CMPIStatus *rc`)
- `CMPIInstance * _makeInst_RouteUsesEndpoint` (const `CMPIBroker * _broker`, const `CMPIObjectPath *ops`, const `CMPIObjectPath *opt`, `CMPIStatus *rc`)
- `CMPIStatus * _assoc_get_NextHopRoute_insts` (const `CMPIBroker * _broker`, const `CMPIContext *ctx`, const `CMPIResult *rslt`, const `CMPIObjectPath *ref`, const `CMPIInstance *sourceInst`, const `char *targetClass`, const `int assocType`, `CMPIStatus *rc`)
- `CMPIStatus * _assoc_get_ProtocolEndpoint_insts` (const `CMPIBroker * _broker`, const `CMPIContext *ctx`, const `CMPIResult *rslt`, const `CMPIObjectPath *ref`, const `CMPIInstance *sourceInst`, const `char *targetClass`, const `int assocType`, `CMPIStatus *rc`)
- `CMPIStatus _assoc_RouteUsesEndpoint` (const `CMPIBroker * _broker`, const `CMPIContext *ctx`, const `CMPIResult *rslt`, const `CMPIObjectPath *ref`, const `char *targetClass`, const `int assocType`, `CMPIStatus *rc`)

### 5.13.1 Detailed Description

`cmpiOSBase_RouteUsesEndpoint.c`

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

**Description:**

This is the factory implementation for creating instances of CIM class Linux\_RouteUsesEndpoint.

Interface Type : Common Magabeability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_RouteUsesEndpoint.c](#).

**5.13.2 Function Documentation**
**5.13.2.1 CMPIStatus\* \_assoc\_get\_NextHopRoute\_insts (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const CMPIInstance \* *sourceInst*, const char \* *targetClass*, int *assocType*, CMPIStatus \* *rc*)**

Retrieve a list of instances from target class (CIM\_NextHopRoute subclass), associated to source class.

**Parameters:**

*\_broker* [in] CIM Object Manager.  
*ctx* [in] context object.  
*rslt* [in] result.  
*ref* [in] source objectPath.  
*sourceInst* [in] source class instance.  
*targetClass* [in] target class name.  
*assocType* [in] association type.  
*rc* [in] cim status.

**Returns:**

cim status.

Definition at line 120 of file cmpiOSBase\_RouteUsesEndpoint.c.

**5.13.2.2 CMPIStatus\* \_assoc\_get\_ProtocolEndpoint\_insts (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const CMPIInstance \* *sourceInst*, const char \* *targetClass*, int *assocType*, CMPIStatus \* *rc*)**

Retrieve a list of instances from target class (CIM\_ProtocolEndpoint subclass), associated to source class.

**Parameters:**

*\_broker* [in] CIM Object Manager.  
*ctx* [in] context object.  
*rslt* [in] result.  
*ref* [in] source objectPath.  
*sourceInst* [in] source class instance.

*targetClass* [in] target class name.

*assocType* [in] association type.

*rc* [in] cim status.

**Returns:**

cim status.

Definition at line 242 of file cmpiOSBase\_RouteUsesEndpoint.c.

**5.13.2.3 CMPIStatus \_assoc\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const char \* *targetClass*, int *assocType*, CMPIStatus \* *rc*)**

Create CMPIInstances of association.

**Parameters:**

*\_broker* [in] CIM Object Manager.

*ctx* [in] context object.

*rslt* [in] result.

*ref* [in] source objectPath.

*targetClass* [in] target class name.

*assocType* [in] association type.

*rc* [in] cim status.

**Returns:**

cim status.

Definition at line 368 of file cmpiOSBase\_RouteUsesEndpoint.c.

**5.13.2.4 CMPIInstance\* \_makeInst\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIObjectPath \* *ops*, const CMPIObjectPath \* *opt*, CMPIStatus \* *rc*)**

Method to create a CMPIInstance of this class.

**Parameters:**

*\_broker* [in] CIM Object Manager.

*ops* [in] Source objectPath.

*opt* [in] Target objectPath.

*rc* [in] cim status.

**Returns:**

created Instance.

Definition at line 79 of file cmpiOSBase\_RouteUsesEndpoint.c.

### 5.13.2.5 CMPIObjectPath\* \_makePath\_RouteUsesEndpoint (const CMPIBroker \* *\_broker*, const CMPIObjectPath \* *ops*, const CMPIObjectPath \* *opt*, CMPIStatus \* *rc*)

Method to create a CMPIObjectPath of this class.

#### Parameters:

*\_broker* [in] CIM Object Manager.  
*ops* [in] Source objectPath.  
*opt* [in] Target objectPath.  
*rc* [in] cim status.

#### Returns:

created ObjectPath.

Definition at line 41 of file cmpiOSBase\_RouteUsesEndpoint.c.



## 5.14 /mnt/storage/TESIS/sblim/cmpi-router/src/cmpiOSBase\_RouteUsesEndpointProvider.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "cmpidt.h"
#include "cmpift.h"
#include "cmpimacs.h"
#include "OSBase_Common.h"
#include "cmpiOSBase_Common.h"
#include "cmpiOSBase_RouteUsesEndpoint.h"
```

### Functions

- `CMPIStatus OSBase_RouteUsesEndpointProviderCleanup` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- `CMPIStatus OSBase_RouteUsesEndpointProviderEnumInstanceNames` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref)
- `CMPIStatus OSBase_RouteUsesEndpointProviderEnumInstances` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref, const char \*\*properties)
- `CMPIStatus OSBase_RouteUsesEndpointProviderGetInstance` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const char \*\*properties)
- `CMPIStatus OSBase_RouteUsesEndpointProviderCreateInstance` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const CMPIInstance \*ci)
- `CMPIStatus OSBase_RouteUsesEndpointProviderSetInstance` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const CMPIInstance \*ci, const char \*\*properties)
- `CMPIStatus OSBase_RouteUsesEndpointProviderDeleteInstance` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop)
- `CMPIStatus OSBase_RouteUsesEndpointProviderExecQuery` (CMPIInstanceMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*ref, const char \*lang, const char \*query)
- `CMPIStatus OSBase_RouteUsesEndpointProviderAssociationCleanup` (CMPIAssociationMI \*mi, const CMPIContext \*ctx, CMPIBoolean terminating)
- `CMPIStatus OSBase_RouteUsesEndpointProviderAssociators` (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const char \*assocClass, const char \*resultClass, const char \*role, const char \*resultRole, const char \*\*propertyList)
- `CMPIStatus OSBase_RouteUsesEndpointProviderAssociatorNames` (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const char \*assocClass, const char \*resultClass, const char \*role, const char \*resultRole)
- `CMPIStatus OSBase_RouteUsesEndpointProviderReferences` (CMPIAssociationMI \*mi, const CMPIContext \*ctx, const CMPIResult \*rslt, const CMPIObjectPath \*cop, const char \*assocClass, const char \*role, const char \*\*propertyList)

- `CMPIStatus OSBase_RouteUsesEndpointProviderReferenceNames` (`CMPIAssociationMI *mi`, `const CMPIContext *ctx`, `const CMPIResult *rslt`, `const CMPIOObjectPath *cop`, `const char *assocClass`, `const char *role`)
- `CMInstanceMIStub` (`OSBase_RouteUsesEndpointProvider`, `OSBase_RouteUsesEndpointProvider`, `_broker`, `CMNoHook`)
- `CMAssociationMIStub` (`OSBase_RouteUsesEndpointProvider`, `OSBase_RouteUsesEndpointProvider`, `_broker`, `CMNoHook`)

## Variables

- `static const CMPIBroker * _broker`

### 5.14.1 Detailed Description

[cmpiOSBase\\_RouteUsesEndpointProvider.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

Linux\_RouteUsesEndpointProvider association provider implementation. The following CMPI instance methods are supported:

- Cleanup
- EnumerateInstanceNames
- EnumerateInstances
- GetInstance
- AssociationCleanup
- Associators
- AssociatorNames
- References
- ReferenceNames

Interface Type : Common Manageability Programming Interface ( CMPI )

Definition in file [cmpiOSBase\\_RouteUsesEndpointProvider.c](#).

### 5.14.2 Function Documentation

**5.14.2.1 CMAssociationMIStub (OSBase\_RouteUsesEndpointProvider, OSBase\_RouteUsesEndpointProvider, \_broker, CMNoHook)**

**5.14.2.2 CMInstanceMIStub (OSBase\_RouteUsesEndpointProvider, OSBase\_RouteUsesEndpointProvider, \_broker, CMNoHook)**

**5.14.2.3 CMPIStatus OSBase\_RouteUsesEndpointProviderAssociationCleanup (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)**

Definition at line 267 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.4 CMPIStatus OSBase\_RouteUsesEndpointProviderAssociatorNames (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *resultClass*, const char \* *role*, const char \* *resultRole*)**

Definition at line 371 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.5 CMPIStatus OSBase\_RouteUsesEndpointProviderAssociators (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *resultClass*, const char \* *role*, const char \* *resultRole*, const char \*\* *propertyList*)**

Definition at line 280 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.6 CMPIStatus OSBase\_RouteUsesEndpointProviderCleanup (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, CMPIBoolean *terminating*)**

Definition at line 59 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.7 CMPIStatus OSBase\_RouteUsesEndpointProviderCreateInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*, const CMPIInstance \* *ci*)**

Definition at line 183 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.8 CMPIStatus OSBase\_RouteUsesEndpointProviderDeleteInstance (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*)**

Definition at line 223 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.9 CMPIStatus OSBase\_RouteUsesEndpointProviderEnumInstanceNames**  
 (CMPIInstanceMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const  
 CMPIObjectPath \* *ref*)

Definition at line 72 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.10 CMPIStatus OSBase\_RouteUsesEndpointProviderEnumInstances** (CMPIInstanceMI \*  
*mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*,  
 const char \*\* *properties*)

Definition at line 109 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.11 CMPIStatus OSBase\_RouteUsesEndpointProviderExecQuery** (CMPIInstanceMI \* *mi*,  
 const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *ref*, const  
 char \* *lang*, const char \* *query*)

Definition at line 241 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.12 CMPIStatus OSBase\_RouteUsesEndpointProviderGetInstance** (CMPIInstanceMI \* *mi*,  
 const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*,  
 const char \*\* *properties*)

Definition at line 146 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.13 CMPIStatus OSBase\_RouteUsesEndpointProviderReferenceNames**  
 (CMPIAssociationMI \* *mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const  
 CMPIObjectPath \* *cop*, const char \* *assocClass*, const char \* *role*)

Definition at line 524 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.14 CMPIStatus OSBase\_RouteUsesEndpointProviderReferences** (CMPIAssociationMI \*  
*mi*, const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*,  
 const char \* *assocClass*, const char \* *role*, const char \*\* *propertyList*)

Definition at line 460 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

**5.14.2.15 CMPIStatus OSBase\_RouteUsesEndpointProviderSetInstance** (CMPIInstanceMI \* *mi*,  
 const CMPIContext \* *ctx*, const CMPIResult \* *rslt*, const CMPIObjectPath \* *cop*,  
 const CMPIInstance \* *ci*, const char \*\* *properties*)

Definition at line 202 of file cmpiOSBase\_RouteUsesEndpointProvider.c.

### 5.14.3 Variable Documentation

#### 5.14.3.1 `const CMPIBroker* _broker` [`static`]

Definition at line 45 of file `cmpiOSBase_RouteUsesEndpointProvider.c`.

## 5.15 /mnt/storage/TESIS/sblim/cmpi-router/src/OSBase\_LANEndpoint.c File Reference

```
#include "OSBase_Common.h"
#include "OSBase_LANEndpoint.h"
#include <unistd.h>
#include <linux/rtnetlink.h>
#include <linux/if.h>
#include <sys/types.h>
#include <signal.h>
```

### Functions

- unsigned long [datetime\\_str\\_interval\\_to\\_ms](#) (const char \*pDatetimeString)
- int [getLANEndpoints](#) (struct [LANEndpointList](#) \*\*lanEPList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [getALLLANEndpoints](#) (struct [LANEndpointList](#) \*\*list)
- int [getLANEndpoint](#) (struct [LANEndpointList](#) \*lanEPList, const char \*linkName)
- unsigned short [changeLinkOPState](#) (const char \*linkName, unsigned short enabledState, unsigned long timeoutPeriod)
- int [nlListTOLanEPList](#) (struct [nlLinkInfoList](#) \*\*nlLinkInfoList, struct [LANEndpointList](#) \*\*lanEPList)
- int [nlInfoTOLanEP](#) (struct [nlLinkInfo](#) \*nlLinkInfo, struct [LANEndpoint](#) \*lanEP)
- void [freeLANEndpointList](#) (struct [LANEndpointList](#) \*lptr)
- void [freeLANEndpoint](#) (struct [LANEndpoint](#) \*sptr)

### 5.15.1 Detailed Description

#### [OSBase\\_LANEndpoint.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This shared library provides resource access functionality for the class `Linux_LANEndpoint`. It is independent from any specific CIM technology.

Definition in file [OSBase\\_LANEndpoint.c](#).

## 5.15.2 Function Documentation

### 5.15.2.1 unsigned short changeLinkOPState (const char \* *linkName*, unsigned short *enabledState*, unsigned long *timeoutPeriod*)

Used to set device status.

#### Parameters:

*linkName* [in] link name.

*enabledState* [in] desired state for the device.

*timeoutPeriod* [in] maximum amount of time that the client expects the transition to the new state to take.

#### Returns:

0 = Completed with No Error 1 = Not Supported 2 = Unknown or Unspecified Error 3 = Cannot complete within Timeout Period 4 = Failed 5 = Invalid Parameter 6 = In Use 7..4095 = DMTF Reserved 4096 = Method Parameters Checked - Job Started 4097 = Invalid State Transition 4098 = Use of Timeout Parameter Not Supported 4099 = Busy 4100..32767 = Method Reserved 32768..65535 = Vendor Specific

Definition at line 157 of file `OSBase_LANEndpoint.c`.

### 5.15.2.2 unsigned long datetime\_str\_interval\_to\_ms (const char \* *pDatetimeString*)

This function converts from datetime string interval to total number of milliseconds (10e-3).

#### Parameters:

*pDatetimeString* [in] datetime string interval to be converted.

#### Returns:

converted datetime string interval.

#### Todo

move this function to the proper file.

Definition at line 38 of file `OSBase_LANEndpoint.c`.

### 5.15.2.3 void freeLANEndpoint (struct LANEndpoint \* *sptr*)

This function is used to clean a [LANEndpoint](#) structure.

**Parameters:**

*sptr* [in] structure to be cleaned.

Definition at line 501 of file OSBase\_LANEndpoint.c.

**5.15.2.4 void freeLANEndpointList (struct LANEndpointList \* *lptr*)**

This function is used to clean a [LANEndpoint](#) list.

**Parameters:**

*lptr* [in] list to be cleaned.

Definition at line 478 of file OSBase\_LANEndpoint.c.

**5.15.2.5 int getALLLANEndpoints (struct LANEndpointList \*\* *list*)**

Definition at line 104 of file OSBase\_LANEndpoint.c.

**5.15.2.6 int getLANEndpoint (struct LANEndpointList \*\* *lanEPList*, const char \* *linkName*)**

This functions is used to get a route base on specified InstanceID.

**Parameters:**

*lanEPList* [out] [LANEndpoint](#) struct to be filled.

*linkName* [in] link ifname.

**Returns:**

0=succesful | 1=fail

Definition at line 130 of file OSBase\_LANEndpoint.c.

**5.15.2.7 int getLANEndpoints (struct LANEndpointList \*\* *lanEPList*, const struct nlLinkInfo \* *nlLinkInfo*)**

This functions is used to get a list of LANEndpoints based on [nlLinkInfo](#) filter.

**Parameters:**

*lanEPList* [out] [LANEndpoint](#) struct to be filled.

*nlLinkInfo* [in] link filter info.

**Returns:**

0=succesful | 1=fail

Definition at line 58 of file OSBase\_LANEndpoint.c.



**5.15.2.8 int nlInfoTOlanEP (struct nlLinkInfo \* *nlLinkInfo*, struct LANEndpoint \* *lanEP*)**

Converts a [nlLinkInfo](#) structure to [LANEndpoint](#) structure.

**Parameters:**

[nlLinkInfo](#) [in] structure to be converted.

*lanEP* [out] converted structure.

**Returns:**

0=succesful | 1=fail

Definition at line 287 of file OSBase\_LANEndpoint.c.

**5.15.2.9 int nlListTOlanEPList (struct nlLinkInfoList \*\* *nlLinkInfoList*, struct LANEndpointList \*\* *lanEPList*)**

Converts a [nlLinkInfoList](#) structure to [LANEndpointList](#) strucutre.

**Parameters:**

[nlLinkInfoList](#) [in] list to be converted.

*lanEPList* [out] converted list.

**Returns:**

0=succesful | 1=fail

Definition at line 244 of file OSBase\_LANEndpoint.c.

## 5.16 /mnt/storage/TESIS/sblim/cmpi-router/src/OSBase\_Netlink.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <linux/netlink.h>
#include <linux/rtnetlink.h>
#include <linux/if_arp.h>
#include <netinet/in.h>
#include <errno.h>
#include <time.h>
#include "OSBase_Common.h"
#include "OSBase_Netlink.h"
```

### Functions

- int [nlOpenSocket](#) ()
- int [nlCloseSocket](#) ()
- int [nlAddAttrToMsg](#) (struct nlmsghdr \*h, int maxlen, int attrType, const void \*attrData, int attrByte-len)
- int [nlAddAttrToMsg32](#) (struct nlmsghdr \*h, int maxlen, int attrType, unsigned int attrData)
- int [nlGetRoutes](#) (struct [nlRouteInfoList](#) \*\*nlRtInfoList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlModifyRoute](#) (struct [nlRouteInfo](#) \*nlRtInfo, int hType, unsigned int hFlags)
- int [nlAddRouteToList](#) (const struct nlmsghdr \*rcvH, struct [nlRouteInfoList](#) \*\*nlRtInfoList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlGenRouteFilter](#) (struct [nlRouteInfo](#) \*nlRtInfo)
- void [nlResetRouteFilter](#) ()
- int [nlCreateDefaultRtInfo](#) (struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlGetLinks](#) (struct [nlLinkInfoList](#) \*\*nlLinkInfoList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [nlModifyLink](#) (struct [nlLinkInfo](#) \*nlLinkInfo, int hType, unsigned int hFlags)
- int [nlAddLinkToList](#) (const struct nlmsghdr \*rcvH, struct [nlLinkInfoList](#) \*\*nlLinkInfoList, const struct [nlLinkInfo](#) \*nlLinkInfo)
- int [nlGenLinkFilter](#) (const struct [nlLinkInfo](#) \*nlLinkInfo)
- void [nlResetLinkFilter](#) ()
- int [nlCreateDefaultLinkInfo](#) (struct [nlLinkInfo](#) \*nlLinkInfo)
- void [nlAddr\\_n2a](#) (const unsigned char \*addr, int alen, int type, char \*buf, int blen)
- unsigned short [nlGetLinkTypePos](#) (unsigned short devType)

### Variables

- struct [nlSockHandle](#) [nlSH](#) = { .fd = -1 }

### 5.16.1 Detailed Description

OSBase\_NetLink.c

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

**Author:**

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

**Contributors:**

**Description:**

Netlink support lib. Based on iproute2 package: (<http://www.linuxfoundation.org/en/Net:Iproute2>)

Definition in file [OSBase\\_Netlink.c](#).

### 5.16.2 Function Documentation

#### 5.16.2.1 `int nlAddAttrToMsg (struct nlmsghdr * h, int maxlen, int attrType, const void * attrData, int attrBytelen)`

This function is used to add an attribute to a netlink message.

**Parameters:**

*h* [out] message.

*maxlen* [in] message length.

*attrType* [in] attribute type.

*attrData* [in] attribute data.

*attrBytelen* [in] attribute length.

**Returns:**

0=succesful | 1=fail

Definition at line 147 of file OSBase\_Netlink.c.

#### 5.16.2.2 `int nlAddAttrToMsg32 (struct nlmsghdr * h, int maxlen, int attrType, unsigned int attrData)`

This function is used to add an attribute to a netlink message.

**Parameters:**

*h* [out] message.  
*maxlen* [in] message length.  
*attrType* [in] attribute type.  
*attrData* [in] attribute data.

**Returns:**

0=succesful | 1=fail

Definition at line 175 of file OSBase\_Netlink.c.

### 5.16.2.3 **int nlAddLinkToList (const struct nlmsg\_hdr \* *rcvH*, struct nlLinkInfoList \*\* *nlLinkInfoList*, const struct nlLinkInfo \* *nlLinkInfo*)**

This function is used to add a link to specified links list.

**Parameters:**

*rcvH* [in] link message.  
*nlLinkInfoList* [out] list where link will be added.  
*nlLinkInfo* [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1490 of file OSBase\_Netlink.c.

### 5.16.2.4 **void nlAddr\_n2a (const unsigned char \* *addr*, int *alen*, int *type*, char \* *buf*, int *blen*)**

Function to convert a L2 address from network representation to string representation (user friendly).

**Parameters:**

*addr* [in] L2 address to convert.  
*alen* [in] L2 address length.  
*type* [in] link type.  
*buf* [out] converted address.  
*blen* [in] converted address maximum length.

Definition at line 1892 of file OSBase\_Netlink.c.

### 5.16.2.5 **int nlAddRouteToList (const struct nlmsg\_hdr \* *rcvH*, struct nlRouteInfoList \*\* *nlRtInfoList*, struct nlRouteInfo \* *nlRtInfo*)**

This function is used to add a route to specified routes list.

**Parameters:**

*rcvH* [in] route message.

*nlRtInfoList* [out] list where route will be added.

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 659 of file OSBase\_Netlink.c.

### 5.16.2.6 int nlCloseSocket ()

Used to close previously open Netlink socket.

**Returns:**

0=succesful | 1=fail

Definition at line 124 of file OSBase\_Netlink.c.

### 5.16.2.7 int nlCreateDefaultLinkInfo (struct nlLinkInfo \* *nlLinkInfo*)

Used to create a [nlLinkInfo](#) structure with default values.

**Parameters:**

[nlLinkInfo](#) [out] link structure.

**Returns:**

0=succesful | 1=fail

Definition at line 1855 of file OSBase\_Netlink.c.

### 5.16.2.8 int nlCreateDefaultRtInfo (struct nlRouteInfo \* *nlRtInfo*)

Used to create a [nlRouteInfo](#) structure with default values.

**Parameters:**

*nlRtInfo* [out] route structure.

**Returns:**

0=succesful | 1=fail

Definition at line 1069 of file OSBase\_Netlink.c.

### 5.16.2.9 int nlGenLinkFilter (const struct nlLinkInfo \* *nlLinkInfo*)

Used to generate a filter to be applied to a list of links.

**Parameters:**

[nlLinkInfo](#) [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1700 of file OSBase\_Netlink.c.

**5.16.2.10 int nlGenRouteFilter (struct nlRouteInfo \* *nlRtInfo*)**

Used to generate a filter to be applied to a list of routes.

**Parameters:**

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 851 of file OSBase\_Netlink.c.

**5.16.2.11 int nlGetLinks (struct nlLinkInfoList \*\* *nlLinkInfoList*, const struct nlLinkInfo \* *nlLinkInfo*)**

This function is used to get links using [nlLinkInfo](#) as filter.

**Parameters:**

[nlLinkInfoList](#) [out] links list (one or more entries).

[nlLinkInfo](#) [in] link info.

**Returns:**

0=succesful | 1=fail

Definition at line 1110 of file OSBase\_Netlink.c.

**5.16.2.12 unsigned short nlGetLinkTypePos (unsigned short *devType*)**

Lookup the position of the passed type into linkType array.

**Parameters:**

*devType* [in] type to find.

**Returns:**

position into linkType array.

Definition at line 1929 of file OSBase\_Netlink.c.

**5.16.2.13 int nlGetRoutes (struct nlRouteInfoList \*\* *nlRtInfoList*, struct nlRouteInfo \* *nlRtInfo*)**

This function is used to get routes using *nlRtInfo* as filter.

**Parameters:**

*nlRtInfoList* [out] routes list (one or more entries).

*nlRtInfo* [in] route info.

**Returns:**

0=succesful | 1=fail

Definition at line 209 of file OSBase\_Netlink.c.

**5.16.2.14 int nlModifyLink (struct nlLinkInfo \* *nlLinkInfo*, int *hType*, unsigned int *hFlags*)**

This function is used to ADD/DELETE/MODIFY a link.

**Parameters:**

*nlLinkInfo* [in] link info.

*hType* [in] type of message.

*hFlags* [in] message flags.

**Returns:**

0=succesful | 1=fail

Definition at line 1279 of file OSBase\_Netlink.c.

**5.16.2.15 int nlModifyRoute (struct nlRouteInfo \* *nlRtInfo*, int *hType*, unsigned int *hFlags*)**

This function is used to ADD/DELETE/MODIFY a route.

**Parameters:**

*nlRtInfo* [in] route info.

*hType* [in] type of message.

*hFlags* [in] message flags.

**Returns:**

0=succesful | 1=fail

Definition at line 384 of file OSBase\_Netlink.c.

**5.16.2.16 int nlOpenSocket ()**

Used to open a Netlink socket.

**Returns:**

0=succesful | 1=fail

Definition at line 48 of file OSBase\_Netlink.c.

**5.16.2.17 void nlResetLinkFilter ()**

Used to reset link filter handler.

Definition at line 1825 of file OSBase\_Netlink.c.

**5.16.2.18 void nlResetRouteFilter ()**

Used to reset route filter handler.

Definition at line 1042 of file OSBase\_Netlink.c.

**5.16.3 Variable Documentation****5.16.3.1 struct nlSockHandle nlSH = { .fd = -1 }**

Netlink socket handler instance.

Definition at line 44 of file OSBase\_Netlink.c.



## 5.17 /mnt/storage/TESIS/sblim/cmpi-router/src/OSBase\_NextHopIPRoute.c File Reference

```
#include "OSBase_Common.h"
#include "OSBase_Netlink.h"
#include "OSBase_NextHopIPRoute.h"
#include <unistd.h>
#include <string.h>
#include <sys/param.h>
#include <linux/rtnetlink.h>
```

### Functions

- int [getIPRoutes](#) (struct [nextHopIPList](#) \*\*nHopIPList, struct [nlRouteInfo](#) \*nlRtInfo)
- int [getAllIPRoutes](#) (struct [nextHopIPList](#) \*\*nHopIPList)
- int [getIPRouteId](#) (struct [nextHopIPList](#) \*\*nHopIPList, char \*instanceId)
- int [addIPRoute](#) (struct [nextHopIP](#) \*nHopIP)
- int [delIPRoute](#) (struct [nextHopIP](#) \*nHopIP)
- int [nlListToNhList](#) (struct [nlRouteInfoList](#) \*\*nlRtInfoList, struct [nextHopIPList](#) \*\*nHopIPList)
- int [nhToNhInfo](#) (struct [nextHopIP](#) \*nHopIP, struct [nlRouteInfo](#) \*nlRtInfo)
- int [nlInfoToNh](#) (struct [nlRouteInfo](#) \*nlRtInfo, struct [nextHopIP](#) \*nHopIP)
- int [getAddrType](#) (const char \*addr)
- int [getAddrTypeStr](#) (int addressType, char \*addrTypeStr)
- void [freeNextHopIPList](#) (struct [nextHopIPList](#) \*lptr)
- void [freeNextHopIP](#) (struct [nextHopIP](#) \*sptr)

### 5.17.1 Detailed Description

#### [OSBase\\_NextHopIPRoute.c](#)

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

This shared library provides resource access functionality for the class `Linux_NextHopIPRoute`. It is independent from any specific CIM technology.

Definition in file [OSBase\\_NextHopIPRoute.c](#).

## 5.17.2 Function Documentation

### 5.17.2.1 `int addIPRoute (struct nextHopIP * nHopIP)`

This function is used to add a new route to FIB.

**Parameters:**

*nHopIP* [in] route to be added.

**Returns:**

0=succesful | 1=fail

Definition at line 143 of file `OSBase_NextHopIPRoute.c`.

### 5.17.2.2 `int delIPRoute (struct nextHopIP * nHopIP)`

This function is used to delete a route from FIB.

**Parameters:**

*nHopIP* [in] route to be deleted.

**Returns:**

0=succesful | 1=fail

Definition at line 186 of file `OSBase_NextHopIPRoute.c`.

### 5.17.2.3 `void freeNextHopIP (struct nextHopIP * sptr)`

This function is used to clean a [nextHopIP](#) structure.

**Parameters:**

*sptr* [in] structure to be cleaned.

Definition at line 583 of file `OSBase_NextHopIPRoute.c`.

### 5.17.2.4 `void freeNextHopIPList (struct nextHopIPList * lptr)`

This function is used to clean a [nextHopIP](#) list.

**Parameters:**

*lptr* [in] list to be cleaned.

Definition at line 560 of file `OSBase_NextHopIPRoute.c`.

#### 5.17.2.5 int getAddrType (const char \* *addr*)

Used to get CIM address type from specified route address.

**Parameters:**

*addr* [int] route address.

**Returns:**

CIM address type.

Definition at line 521 of file OSBase\_NextHopIPRoute.c.

#### 5.17.2.6 int getAddrTypeStr (int *addressType*, char \* *addrTypeStr*)

Used to get CIM address type in string format.

**Parameters:**

*addressType* [in] address type identifier.

*addrTypeStr* [out] address type string.

**Returns:**

0=succesful | 1=fail

Definition at line 540 of file OSBase\_NextHopIPRoute.c.

#### 5.17.2.7 int getAllIPRoutes (struct nextHopIPList \*\* *nHopIPList*)

This functions is used to get all ipv4 and ipv6 routes from fib.

**Parameters:**

*nHopIPList* [out] routes struct to be filled.

**Returns:**

0=succesful | 1=fail

Definition at line 77 of file OSBase\_NextHopIPRoute.c.

#### 5.17.2.8 int getIPRouteId (struct nextHopIPList \*\* *nHopIPList*, char \* *instanceId*)

This functions is used to get a route base on specified InstanceID.

**Parameters:**

*nHopIPList* [out] routes struct to be filled.

*instanceId* [in] route InstanceID.

**Returns:**

0=succesful | 1=fail

Definition at line 110 of file OSBase\_NextHopIPRoute.c.

### 5.17.2.9 int getIPRoutes (struct nextHopIPList \*\* *nHopIPList*, struct nlRouteInfo \* *nlRtInfo*)

This functions is used to get a list of routes based on *nlRtInfo* filter.

#### Parameters:

*nHopIPList* [out] routes struct to be filled.

*nlRtInfo* [in] route info.

#### Returns:

0=succesful | 1=fail

Definition at line 31 of file OSBase\_NextHopIPRoute.c.

### 5.17.2.10 int nhTOnlInfo (struct nextHopIP \* *nHopIP*, struct nlRouteInfo \* *nlRtInfo*)

Converts a [nextHopIP](#) structure to [nlRouteInfo](#) structure.

#### Parameters:

*nHopIP* [in] structure to be converted.

*nlRtInfo* [out] converted structure.

#### Returns:

0=succesful | 1=fail

Definition at line 273 of file OSBase\_NextHopIPRoute.c.

### 5.17.2.11 int nlInfoTOnh (struct nlRouteInfo \* *nlRtInfo*, struct nextHopIP \* *nHopIP*)

Converts a [nlRouteInfo](#) structure to [nextHopIP](#) structure.

#### Parameters:

*nlRtInfo* [in] structure to be converted.

*nHopIP* [out] converted structure.

#### Returns:

0=succesful | 1=fail

Definition at line 381 of file OSBase\_NextHopIPRoute.c.

### 5.17.2.12 int nlListTOnhList (struct nlRouteInfoList \*\* *nlRtInfoList*, struct nextHopIPList \*\* *nHopIPList*)

Converts a [nlRouteInfoList](#) structure to [nextHopIPList](#) strucutre.

#### Parameters:

*nlRtInfoList* [in] list to be converted.

*nHopIPList* [out] converted list.

**Returns:**

0=succesful | 1=fail

Definition at line 233 of file OSBase\_NextHopIPRoute.c.

## 5.18 /mnt/storage/TESIS/sblim/mpi-router/src/OSBase\_Zebra.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include "OSBase_Common.h"
#include "OSBase_Netlink.h"
```

### 5.18.1 Detailed Description

OSBase\_NetLink.c

THIS FILE IS PROVIDED UNDER THE TERMS OF THE ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THIS FILE CONSTITUTES RECIPIENTS ACCEPTANCE OF THE AGREEMENT.

You can obtain a current copy of the Eclipse Public License from <http://www.opensource.org/licenses/eclipse-1.0.php>

#### Author:

Federico Martin Casares ([warptrosse@gmail.com](mailto:warptrosse@gmail.com))

#### Contributors:

#### Description:

#### Todo

WILL BE USED IN THE FUTURE TO OBTAIN INFO FROM ZEBRA/QUAGGA.

Definition in file [OSBase\\_Zebra.c](#).

# Index

/mnt/storage/TEStIS/sblim/mpi-  
router/include/OSBase\_LANEndpoint.h, 37

/mnt/storage/TEStIS/sblim/mpi-  
router/include/OSBase\_Netlink.h, 47

/mnt/storage/TEStIS/sblim/mpi-  
router/include/OSBase\_-  
NextHopIPRoute.h, 59

/mnt/storage/TEStIS/sblim/mpi-  
router/include/OSBase\_Zebra.h, 67

/mnt/storage/TEStIS/sblim/mpi-  
router/include/mpiOSBase\_-  
LANEndpoint.h, 27

/mnt/storage/TEStIS/sblim/mpi-  
router/include/mpiOSBase\_-  
NextHopIPRoute.h, 30

/mnt/storage/TEStIS/sblim/mpi-  
router/include/mpiOSBase\_-  
RouteUsesEndpoint.h, 32

/mnt/storage/TEStIS/sblim/mpi-  
router/src/OSBase\_LANEndpoint.c, 94

/mnt/storage/TEStIS/sblim/mpi-  
router/src/OSBase\_Netlink.c, 98

/mnt/storage/TEStIS/sblim/mpi-  
router/src/OSBase\_NextHopIPRoute.c, 105

/mnt/storage/TEStIS/sblim/mpi-  
router/src/OSBase\_Zebra.c, 110

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
CSHostedRouteProvider.c, 68

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_LANEndpoint.c, 73

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
LANEndpointProvider.c, 75

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
NextHopIPRoute.c, 79

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
NextHopIPRouteProvider.c, 81

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
RouteUsesEndpoint.c, 85

/mnt/storage/TEStIS/sblim/mpi-  
router/src/mpiOSBase\_-  
RouteUsesEndpointProvider.c, 89

\_ClassName  
    mpiOSBase\_CSHostedRouteProvider.c, 71  
    mpiOSBase\_LANEndpoint.h, 29  
    mpiOSBase\_NextHopIPRoute.h, 31  
    mpiOSBase\_RouteUsesEndpoint.h, 35

\_RefLeft  
    mpiOSBase\_CSHostedRouteProvider.c, 72  
    mpiOSBase\_RouteUsesEndpoint.h, 35

\_RefLeftClass  
    mpiOSBase\_CSHostedRouteProvider.c, 72  
    mpiOSBase\_RouteUsesEndpoint.h, 35

\_RefLeftClasses  
    mpiOSBase\_RouteUsesEndpoint.h, 35

\_RefRight  
    mpiOSBase\_CSHostedRouteProvider.c, 72  
    mpiOSBase\_RouteUsesEndpoint.h, 36

\_RefRightClass  
    mpiOSBase\_CSHostedRouteProvider.c, 72  
    mpiOSBase\_RouteUsesEndpoint.h, 36

\_RefRightClasses  
    mpiOSBase\_RouteUsesEndpoint.h, 36

\_assoc\_RouteUsesEndpoint  
    mpiOSBase\_RouteUsesEndpoint.c, 87  
    mpiOSBase\_RouteUsesEndpoint.h, 34

\_assoc\_get\_NextHopRoute\_insts  
    mpiOSBase\_RouteUsesEndpoint.c, 86  
    mpiOSBase\_RouteUsesEndpoint.h, 33

\_assoc\_get\_ProtocolEndpoint\_insts  
    mpiOSBase\_RouteUsesEndpoint.c, 86  
    mpiOSBase\_RouteUsesEndpoint.h, 34

\_broker  
    mpiOSBase\_CSHostedRouteProvider.c, 71  
    mpiOSBase\_LANEndpointProvider.c, 78  
    mpiOSBase\_NextHopIPRouteProvider.c, 84  
    mpiOSBase\_RouteUsesEndpointProvider.c, 92

\_makeInst\_LANEndpoint  
    mpiOSBase\_LANEndpoint.c, 74  
    mpiOSBase\_LANEndpoint.h, 28

\_makeInst\_NextHopIPRoute

- cmpiOSBase\_NextHopIPRoute.c, 80
  - cmpiOSBase\_NextHopIPRoute.h, 30
- \_makeInst\_RouteUsesEndpoint
  - cmpiOSBase\_RouteUsesEndpoint.c, 87
  - cmpiOSBase\_RouteUsesEndpoint.h, 34
- \_makePath\_LANEndpoint
  - cmpiOSBase\_LANEndpoint.c, 74
  - cmpiOSBase\_LANEndpoint.h, 28
- \_makePath\_NextHopIPRoute
  - cmpiOSBase\_NextHopIPRoute.c, 80
  - cmpiOSBase\_NextHopIPRoute.h, 31
- \_makePath\_RouteUsesEndpoint
  - cmpiOSBase\_RouteUsesEndpoint.c, 87
  - cmpiOSBase\_RouteUsesEndpoint.h, 35
- addIPRoute
  - OSBase\_NextHopIPRoute.c, 106
  - OSBase\_NextHopIPRoute.h, 63
- address
  - nlLinkInfo, 17
  - OSBase\_Netlink.h, 54
- addressLen
  - nlLinkInfo, 17
- addressType
  - nextHopIP, 13
- adminDistance
  - nextHopIP, 13
- aliasAddresses
  - LANEndpoint, 8
- ARRAY\_SIZE
  - OSBase\_Netlink.h, 49
- ATYPE\_ASSOC
  - cmpiOSBase\_RouteUsesEndpoint.h, 33
- ATYPE ASSO CN
  - cmpiOSBase\_RouteUsesEndpoint.h, 33
- ATYPE\_REFER
  - cmpiOSBase\_RouteUsesEndpoint.h, 33
- ATYPE\_REFER N
  - cmpiOSBase\_RouteUsesEndpoint.h, 33
- availReqStates
  - LANEndpoint, 8
- broadcast
  - nlLinkInfo, 17
  - OSBase\_Netlink.h, 54
- broadcastLen
  - nlLinkInfo, 18
- caption
  - LANEndpoint, 8
  - nextHopIP, 13
- change
  - nlLinkInfo, 18
  - OSBase\_Netlink.h, 55
- changeLinkOPState
  - OSBase\_LANEndpoint.c, 95
  - OSBase\_LANEndpoint.h, 44
- CMAssociationMISub
  - cmpiOSBase\_CSHostedRouteProvider.c, 70
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 90
- CMInstanceMISub
  - cmpiOSBase\_CSHostedRouteProvider.c, 70
  - cmpiOSBase\_LANEndpointProvider.c, 76
  - cmpiOSBase\_NextHopIPRouteProvider.c, 82
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- CMMMethodMISub
  - cmpiOSBase\_LANEndpointProvider.c, 76
  - cmpiOSBase\_NextHopIPRouteProvider.c, 82
- cmpiOSBase\_RouteUsesEndpoint.h
  - ATYPE\_ASSOC, 33
  - ATYPE ASSO CN, 33
  - ATYPE\_REFER, 33
  - ATYPE\_REFER N, 33
- cmpiOSBase\_CSHostedRouteProvider.c
  - \_ClassName, 71
  - \_RefLeft, 72
  - \_RefLeftClass, 72
  - \_RefRight, 72
  - \_RefRightClass, 72
  - \_broker, 71
  - CMAssociationMISub, 70
  - CMInstanceMISub, 70
  - OSBase\_CSHostedRouteProviderAssociationCleanup, 70
  - OSBase\_CSHostedRouteProviderAssociatorNames, 70
  - OSBase\_CSHostedRouteProviderAssociators, 70
  - OSBase\_CSHostedRouteProviderCleanup, 70
  - OSBase\_CSHostedRouteProviderCreateInstance, 70
  - OSBase\_CSHostedRouteProviderDeleteInstance, 70
  - OSBase\_CSHostedRouteProviderEnumInstanceNames, 70
  - OSBase\_CSHostedRouteProviderEnumInstances, 71
  - OSBase\_CSHostedRouteProviderExecQuery, 71
  - OSBase\_CSHostedRouteProviderGetInstance, 71
  - OSBase\_CSHostedRouteProviderReferenceNames, 71
  - OSBase\_CSHostedRouteProviderReferences, 71



- OSBase\_CSHostedRouteProviderSetInstance, 71
- cmpiOSBase\_LANEndpoint.c
  - \_makeInst\_LANEndpoint, 74
  - \_makePath\_LANEndpoint, 74
- cmpiOSBase\_LANEndpoint.h
  - \_ClassName, 29
  - \_makeInst\_LANEndpoint, 28
  - \_makePath\_LANEndpoint, 28
- cmpiOSBase\_LANEndpointProvider.c
  - \_broker, 78
  - CMInstanceMISub, 76
  - CMMMethodMISub, 76
  - OSBase\_LANEndpointProviderCleanup, 76
  - OSBase\_LANEndpointProviderCreateInstance, 77
  - OSBase\_LANEndpointProviderDeleteInstance, 77
  - OSBase\_LANEndpointProviderEnumInstanceNames, 77
  - OSBase\_LANEndpointProviderEnumInstances, 77
  - OSBase\_LANEndpointProviderExecQuery, 77
  - OSBase\_LANEndpointProviderGetInstance, 77
  - OSBase\_LANEndpointProviderInvokeMethod, 77
  - OSBase\_LANEndpointProviderMethodCleanup, 77
  - OSBase\_LANEndpointProviderSetInstance, 78
- cmpiOSBase\_NextHopIPRoute.c
  - \_makeInst\_NextHopIPRoute, 80
  - \_makePath\_NextHopIPRoute, 80
- cmpiOSBase\_NextHopIPRoute.h
  - \_ClassName, 31
  - \_makeInst\_NextHopIPRoute, 30
  - \_makePath\_NextHopIPRoute, 31
- cmpiOSBase\_NextHopIPRouteProvider.c
  - \_broker, 84
  - CMInstanceMISub, 82
  - CMMMethodMISub, 82
  - getNextHopIPParams, 82
  - OSBase\_NextHopIPRouteProviderCleanup, 83
  - OSBase\_NextHopIPRouteProviderCreateInstance, 83
  - OSBase\_NextHopIPRouteProviderDeleteInstance, 83
  - OSBase\_NextHopIPRouteProviderEnumInstanceNames, 83
  - OSBase\_NextHopIPRouteProviderEnumInstances, 83
- OSBase\_NextHopIPRouteProviderExecQuery, 83
- OSBase\_NextHopIPRouteProviderGetInstance, 83
- OSBase\_NextHopIPRouteProviderInvokeMethod, 84
- OSBase\_NextHopIPRouteProviderMethodCleanup, 84
- OSBase\_NextHopIPRouteProviderSetInstance, 84
- cmpiOSBase\_RouteUsesEndpoint.c
  - \_assoc\_RouteUsesEndpoint, 87
  - \_assoc\_get\_NextHopRoute\_insts, 86
  - \_assoc\_get\_ProtocolEndpoint\_insts, 86
  - \_makeInst\_RouteUsesEndpoint, 87
  - \_makePath\_RouteUsesEndpoint, 87
- cmpiOSBase\_RouteUsesEndpoint.h
  - \_ClassName, 35
  - \_RefLeft, 35
  - \_RefLeftClass, 35
  - \_RefLeftClasses, 35
  - \_RefRight, 36
  - \_RefRightClass, 36
  - \_RefRightClasses, 36
  - \_assoc\_RouteUsesEndpoint, 34
  - \_assoc\_get\_NextHopRoute\_insts, 33
  - \_assoc\_get\_ProtocolEndpoint\_insts, 34
  - \_makeInst\_RouteUsesEndpoint, 34
  - \_makePath\_RouteUsesEndpoint, 35
- cmpiOSBase\_RouteUsesEndpointProvider.c
  - \_broker, 92
  - CMAssociationMISub, 90
  - CMInstanceMISub, 91
  - OSBase\_RouteUsesEndpointProviderAssociationCleanup, 91
  - OSBase\_RouteUsesEndpointProviderAssociatorNames, 91
  - OSBase\_RouteUsesEndpointProviderAssociators, 91
  - OSBase\_RouteUsesEndpointProviderCleanup, 91
  - OSBase\_RouteUsesEndpointProviderCreateInstance, 91
  - OSBase\_RouteUsesEndpointProviderDeleteInstance, 91
  - OSBase\_RouteUsesEndpointProviderEnumInstanceNames, 91
  - OSBase\_RouteUsesEndpointProviderEnumInstances, 92
  - OSBase\_RouteUsesEndpointProviderExecQuery, 92
  - OSBase\_RouteUsesEndpointProviderGetInstance, 92

- OSBase\_RouteUsesEndpointProviderReferenceNames, OSBase\_LANEndpoint.c, 95
- 92 OSBase\_LANEndpoint.h, 44
- OSBase\_RouteUsesEndpointProviderReferences, freeLANEndpointList
- 92 OSBase\_LANEndpoint.c, 96
- OSBase\_RouteUsesEndpointProviderSetInstance, OSBase\_LANEndpoint.h, 44
- 92
- freeNextHopIP
- OSBase\_NextHopIPRoute.c, 106
- OSBase\_NextHopIPRoute.h, 63
- freeNextHopIPList
- OSBase\_NextHopIPRoute.c, 106
- OSBase\_NextHopIPRoute.h, 63
- getAddrType
- OSBase\_NextHopIPRoute.c, 106
- OSBase\_NextHopIPRoute.h, 63
- getAddrTypeStr
- OSBase\_NextHopIPRoute.c, 107
- OSBase\_NextHopIPRoute.h, 64
- getAllIPRoutes
- OSBase\_NextHopIPRoute.c, 107
- OSBase\_NextHopIPRoute.h, 64
- getALLLANEndpoint
- OSBase\_LANEndpoint.h, 44
- getALLLANEndpoints
- OSBase\_LANEndpoint.c, 96
- getIPRouteId
- OSBase\_NextHopIPRoute.c, 107
- OSBase\_NextHopIPRoute.h, 64
- getIPRoutes
- OSBase\_NextHopIPRoute.c, 107
- OSBase\_NextHopIPRoute.h, 64
- getLANEndpoint
- OSBase\_LANEndpoint.c, 96
- OSBase\_LANEndpoint.h, 45
- getLANEndpoints
- OSBase\_LANEndpoint.c, 96
- OSBase\_LANEndpoint.h, 45
- getNextHopIPParams
- cmpiOSBase\_NextHopIPRouteProvider.c, 82
- groupAddresses
- LANEndpoint, 9
- gw
- nlRouteInfo, 22
- OSBase\_Netlink.h, 55
- healthState
- LANEndpoint, 9
- ifname
- nlLinkInfo, 18
- OSBase\_Netlink.h, 55
- index
- nlLinkInfo, 18
- OSBase\_Netlink.h, 55
- OSBase\_RouteUsesEndpointProviderReferenceNames, OSBase\_LANEndpoint.c, 95
- 92 OSBase\_LANEndpoint.h, 44
- OSBase\_RouteUsesEndpointProviderReferences, freeLANEndpointList
- 92 OSBase\_LANEndpoint.c, 96
- OSBase\_RouteUsesEndpointProviderSetInstance, OSBase\_LANEndpoint.h, 44
- 92
- communicationStatus
- LANEndpoint, 8
- CREATION\_CLASS\_NAME
- OSBase\_LANEndpoint.h, 39
- creationClassName
- LANEndpoint, 8
- datetime\_str\_interval\_to\_ms
- OSBase\_LANEndpoint.c, 95
- delIPRoute
- OSBase\_NextHopIPRoute.c, 106
- OSBase\_NextHopIPRoute.h, 63
- description
- LANEndpoint, 8
- nextHopIP, 14
- detailedStatus
- LANEndpoint, 9
- dstAddr
- nlRouteInfo, 21
- OSBase\_Netlink.h, 55
- dstAddress
- nextHopIP, 14
- dstLen
- nlRouteInfo, 21
- OSBase\_Netlink.h, 55
- dstMask
- nextHopIP, 14
- elementName
- LANEndpoint, 9
- nextHopIP, 14
- enabledDefault
- LANEndpoint, 9
- enabledState
- LANEndpoint, 9
- family
- nlLinkInfo, 18
- nlRouteInfo, 21
- OSBase\_Netlink.h, 55
- fd
- nlSockHandle, 25
- flags
- nlLinkInfo, 18
- OSBase\_Netlink.h, 55
- FREE\_SAFE
- OSBase\_Netlink.h, 49
- freeLANEndpoint

- inputIf
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 55
- instanceID
  - LANEndpoint, 9
  - nextHopIP, 14
- INSTANCEID\_FORMAT
  - OSBase\_NextHopIPRoute.h, 60
- INSTANCEID\_FORMAT\_PARSE
  - OSBase\_NextHopIPRoute.h, 60
- isStatic
  - nextHopIP, 14
- LANEndpoint, 7
  - aliasAddresses, 8
  - availReqStates, 8
  - caption, 8
  - communicationStatus, 8
  - creationClassName, 8
  - description, 8
  - detailedStatus, 9
  - elementName, 9
  - enabledDefault, 9
  - enabledState, 9
  - groupAddresses, 9
  - healthState, 9
  - instanceID, 9
  - lanID, 9
  - macAddress, 9
  - maxDataSize, 10
  - name, 10
  - nameFormat, 10
  - operatingStatus, 10
  - operationalStatus, 10
  - otherEnabledState, 10
  - otherTypeDescription, 10
  - primaryStatus, 10
  - protocolIFType, 10
  - requestedState, 11
  - statusDescriptions, 11
  - systemCreationClassName, 11
  - systemName, 11
  - transitioningToState, 11
- LANENDPOINT\_CAPTION
  - OSBase\_LANEndpoint.h, 39
- LANENDPOINT\_DESC
  - OSBase\_LANEndpoint.h, 39
- LANEndpointList, 12
  - next, 12
  - sptr, 12
- lanID
  - LANEndpoint, 9
- LEP\_CS\_COMMUNICATION\_OK
  - OSBase\_LANEndpoint.h, 41
- LEP\_CS\_LOST\_COMMUNICATION
  - OSBase\_LANEndpoint.h, 41
- LEP\_CS\_NO\_CONTACT
  - OSBase\_LANEndpoint.h, 41
- LEP\_CS\_NOT\_AVAILABLE
  - OSBase\_LANEndpoint.h, 41
- LEP\_CS\_UNKNOWN
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_NO\_ADDITIONAL\_INFORMATION
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_NON\_RECOVERABLE\_ERROR
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_NOT\_AVAILABLE
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_PREDICTIVE\_FAILURE
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_STRESSED
  - OSBase\_LANEndpoint.h, 41
- LEP\_DS\_SUPPORTING\_ENTITY\_IN\_ERROR
  - OSBase\_LANEndpoint.h, 41
- LEP\_ED\_DISABLED
  - OSBase\_LANEndpoint.h, 40
- LEP\_ED\_ENABLED
  - OSBase\_LANEndpoint.h, 40
- LEP\_ED\_ENABLED\_BUT\_OFFLINE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ED\_NO\_DEFAULT
  - OSBase\_LANEndpoint.h, 40
- LEP\_ED\_NOT\_APPLICABLE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ED\_QUIESCE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_DEFERRED
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_DISABLED
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_ENABLED
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_ENABLED\_BUT\_OFFLINE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_IN\_TEST
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_NOT\_APPLICABLE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_OTHER
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_QUIESCE
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_SHUTTING\_DOWN
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_STARTING
  - OSBase\_LANEndpoint.h, 40
- LEP\_ES\_UNKNOWN
  - OSBase\_LANEndpoint.h, 40

- LEP\_HS\_CRITICAL\_FAILURE
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_DEGRADED\_WARNING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_MAJOR\_FAILURE
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_MINOR\_FAILURE
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_NON\_RECOVERABLE\_ERROR
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_OK
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_HS\_UNKNOWN
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OPS\_ABORTED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_COMPLETED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_DEGRADED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_DORMANT
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_ERROR
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_IN\_SERVICE
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_LOST\_COMMUNICATION
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_NO\_CONTACT
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_NON\_RECOVERABLE\_ERROR
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_OK
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_OTHER
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_POWER\_MODE
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_PREDICTIVE\_FAILURE
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_STARTING
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_STOPPED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_STOPPING
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_STRESSED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_SUPPORTING\_ENTITY\_IN\_ERROR
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OPS\_UNKNOWN
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_OS\_ABORTED
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_COMPLETED
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_DORMANT
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_EMIGRATING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_IMMIGRATING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_IN\_SERVICE
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_IN\_TEST
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_MIGRATING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_NOT\_AVAILABLE
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_SERVICING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_SHUTTING\_DOWN
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_SNAPSHOTTING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_STARTING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_STOPPED
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_STOPPING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_TRANSITIONING
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_OS\_UNKNOWN
  - OSBase\_LANEndpoint.h, [42](#)
- LEP\_PS\_DEGRADED
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_PS\_ERROR
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_PS\_OK
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_PS\_UNKNOWN
  - OSBase\_LANEndpoint.h, [43](#)
- LEP\_RS\_DEFERRED
  - OSBase\_LANEndpoint.h, [41](#)
- LEP\_RS\_DISABLED
  - OSBase\_LANEndpoint.h, [40](#)
- LEP\_RS\_ENABLED
  - OSBase\_LANEndpoint.h, [40](#)
- LEP\_RS\_NO\_CHANGE
  - OSBase\_LANEndpoint.h, [41](#)
- LEP\_RS\_NOT\_APPLICABLE
  - OSBase\_LANEndpoint.h, [41](#)
- LEP\_RS\_OFFLINE
  - OSBase\_LANEndpoint.h, [41](#)
- LEP\_RS\_QUIESCE
  - OSBase\_LANEndpoint.h, [41](#)

- LEP\_RS\_REBOOT
  - OSBase\_LANEndpoint.h, 41
- LEP\_RS\_RESET
  - OSBase\_LANEndpoint.h, 41
- LEP\_RS\_SHUT\_DOWN
  - OSBase\_LANEndpoint.h, 40
- LEP\_RS\_TEST
  - OSBase\_LANEndpoint.h, 41
- LEP\_RS\_UNKNOWN
  - OSBase\_LANEndpoint.h, 40
- link
  - nlLinkInfo, 18
  - OSBase\_Netlink.h, 55
- linkFlt
  - OSBase\_Netlink.h, 55
- linkmode
  - nlLinkInfo, 18
  - OSBase\_Netlink.h, 56
- linkType
  - OSBase\_Netlink.h, 56
- linkTypeName
  - OSBase\_Netlink.h, 56
- local
  - nlSockHandle, 25
- macAddress
  - LANEndpoint, 9
- map
  - nlLinkInfo, 18
  - OSBase\_Netlink.h, 56
- maxDataSize
  - LANEndpoint, 10
- MAXHOSTNAMELEN
  - OSBase\_LANEndpoint.h, 39
- metrics
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- mtu
  - nlLinkInfo, 19
  - OSBase\_Netlink.h, 57
- name
  - LANEndpoint, 10
- nameFormat
  - LANEndpoint, 10
- next
  - LANEndpointList, 12
  - nextHopIPList, 16
  - nlLinkInfoList, 20
  - nlRouteInfoList, 24
- nextHopIP, 13
  - addressType, 13
  - adminDistance, 13
  - caption, 13
  - description, 14
  - dstAddress, 14
  - dstMask, 14
  - elementName, 14
  - instanceID, 14
  - isStatic, 14
  - otherDerivation, 14
  - prefixLength, 14
  - routeDerivation, 14
  - routeGateway, 15
  - routeMetric, 15
  - routeOutputIf, 15
  - routeScope, 15
  - routeTable, 15
  - routeType, 15
  - typeOfRoute, 15
- nextHopIPList, 16
  - next, 16
  - sptr, 16
- NH\_AT\_IPV4
  - OSBase\_NextHopIPRoute.h, 61
- NH\_AT\_IPV6
  - OSBase\_NextHopIPRoute.h, 61
- NH\_AT\_UNKNOWN
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_BGP
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_CONNECTED
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_EGP
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_EIGRP
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_HELLO
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_IGRP
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_ISIS
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_OSPF
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_OTHER
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_RIP
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_UNKNOWN
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTD\_USER\_DEFINED
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTS\_HOST
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTS\_LINK
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTS\_NOWHERE

- OSBase\_NextHopIPRoute.h, 61
- NH\_RTS\_SITE
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTS\_UNIVERSE
  - OSBase\_NextHopIPRoute.h, 61
- NH\_RTT\_COMPAT
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTT\_DEFAULT
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTT\_LOCAL
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTT\_MAIN
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTT\_UNSPEC
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_ANYCAST
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_BLACKHOLE
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_BROADCAST
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_LOCAL
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_MULTICAST
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_NAT
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_PROHIBIT
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_THROW
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_UNICAST
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_UNREACHABLE
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_UNSPEC
  - OSBase\_NextHopIPRoute.h, 62
- NH\_RTTY\_XRESOLVE
  - OSBase\_NextHopIPRoute.h, 62
- NH\_TOR\_ACTUAL
  - OSBase\_NextHopIPRoute.h, 62
- NH\_TOR\_ADMINISTRATOR
  - OSBase\_NextHopIPRoute.h, 62
- NH\_TOR\_COMPUTED
  - OSBase\_NextHopIPRoute.h, 62
- nhTonlInfo
  - OSBase\_NextHopIPRoute.c, 108
  - OSBase\_NextHopIPRoute.h, 65
- NL\_SOCK\_RCV\_BUFF\_LEN
  - OSBase\_Netlink.h, 49
- NL\_SOCK\_SND\_BUFF\_LEN
  - OSBase\_Netlink.h, 49
- nlAddAttrToMsg
  - OSBase\_Netlink.c, 99
- OSBase\_Netlink.h, 50
- nlAddAttrToMsg32
  - OSBase\_Netlink.c, 99
  - OSBase\_Netlink.h, 50
- nlAddLinkToList
  - OSBase\_Netlink.c, 100
  - OSBase\_Netlink.h, 50
- nlAddr\_n2a
  - OSBase\_Netlink.c, 100
  - OSBase\_Netlink.h, 51
- nlAddRouteToList
  - OSBase\_Netlink.c, 100
  - OSBase\_Netlink.h, 51
- nlCloseSocket
  - OSBase\_Netlink.c, 101
  - OSBase\_Netlink.h, 51
- nlCreateDefaultLinkInfo
  - OSBase\_Netlink.c, 101
  - OSBase\_Netlink.h, 51
- nlCreateDefaultRtInfo
  - OSBase\_Netlink.c, 101
  - OSBase\_Netlink.h, 52
- nlGenLinkFilter
  - OSBase\_Netlink.c, 101
  - OSBase\_Netlink.h, 52
- nlGenRouteFilter
  - OSBase\_Netlink.c, 102
  - OSBase\_Netlink.h, 52
- nlGetLinks
  - OSBase\_Netlink.c, 102
  - OSBase\_Netlink.h, 52
- nlGetLinkTypePos
  - OSBase\_Netlink.c, 102
  - OSBase\_Netlink.h, 53
- nlGetRoutes
  - OSBase\_Netlink.c, 102
  - OSBase\_Netlink.h, 53
- nlInfoTOLanEP
  - OSBase\_LANEndpoint.c, 96
  - OSBase\_LANEndpoint.h, 45
- nlInfoTOnh
  - OSBase\_NextHopIPRoute.c, 108
  - OSBase\_NextHopIPRoute.h, 65
- nlLinkInfo, 17
  - address, 17
  - addressLen, 17
  - broadcast, 17
  - broadcastLen, 18
  - change, 18
  - family, 18
  - flags, 18
  - ifname, 18
  - index, 18
  - link, 18

- linkmode, 18
- map, 18
- mtu, 19
- operstate, 19
- qdisc, 19
- stats, 19
- txqlen, 19
- type, 19
- nlLinkInfoList, 20
  - next, 20
  - sptr, 20
- nlListTOLanEPList
  - OSBase\_LANEndpoint.c, 97
  - OSBase\_LANEndpoint.h, 45
- nlListTONhList
  - OSBase\_NextHopIPRoute.c, 108
  - OSBase\_NextHopIPRoute.h, 65
- nlModifyLink
  - OSBase\_Netlink.c, 103
  - OSBase\_Netlink.h, 53
- nlModifyRoute
  - OSBase\_Netlink.c, 103
  - OSBase\_Netlink.h, 54
- NLMSG\_TAIL
  - OSBase\_Netlink.h, 49
- nlOpenSocket
  - OSBase\_Netlink.c, 103
  - OSBase\_Netlink.h, 54
- nlResetLinkFilter
  - OSBase\_Netlink.c, 103
  - OSBase\_Netlink.h, 54
- nlResetRouteFilter
  - OSBase\_Netlink.c, 104
  - OSBase\_Netlink.h, 54
- nlRouteInfo, 21
  - dstAddr, 21
  - dstLen, 21
  - family, 21
  - gw, 22
  - inputIf, 22
  - metrics, 22
  - outputIf, 22
  - prefSrc, 22
  - priority, 22
  - protocol, 22
  - scope, 22
  - srcAddr, 22
  - srcLen, 23
  - table, 23
  - tos, 23
  - type, 23
- nlRouteInfoList, 24
  - next, 24
  - sptr, 24
- nlSH
  - OSBase\_Netlink.c, 104
- nlSockHandle, 25
  - fd, 25
  - local, 25
- operatingStatus
  - LANEndpoint, 10
- operationalStatus
  - LANEndpoint, 10
- operstate
  - nlLinkInfo, 19
  - OSBase\_Netlink.h, 57
- OSBase\_LANEndpoint.h
  - LEP\_CS\_COMMUNICATION\_OK, 41
  - LEP\_CS\_LOST\_COMMUNICATION, 41
  - LEP\_CS\_NO\_CONTACT, 41
  - LEP\_CS\_NOT\_AVAILABLE, 41
  - LEP\_CS\_UNKNOWN, 41
  - LEP\_DS\_NO\_ADDITIONAL\_INFORMATION, 41
  - LEP\_DS\_NON\_RECOVERABLE\_ERROR, 41
  - LEP\_DS\_NOT\_AVAILABLE, 41
  - LEP\_DS\_PREDICTIVE\_FAILURE, 41
  - LEP\_DS\_STRESSED, 41
  - LEP\_DS\_SUPPORTING\_ENTITY\_IN\_ERROR, 41
  - LEP\_ED\_DISABLED, 40
  - LEP\_ED\_ENABLED, 40
  - LEP\_ED\_ENABLED\_BUT\_OFFLINE, 40
  - LEP\_ED\_NO\_DEFAULT, 40
  - LEP\_ED\_NOT\_APPLICABLE, 40
  - LEP\_ED QUIESCE, 40
  - LEP\_ES\_DEFERRED, 40
  - LEP\_ES\_DISABLED, 40
  - LEP\_ES\_ENABLED, 40
  - LEP\_ES\_ENABLED\_BUT\_OFFLINE, 40
  - LEP\_ES\_IN\_TEST, 40
  - LEP\_ES\_NOT\_APPLICABLE, 40
  - LEP\_ES\_OTHER, 40
  - LEP\_ES QUIESCE, 40
  - LEP\_ES\_SHUTTING\_DOWN, 40
  - LEP\_ES\_STARTING, 40
  - LEP\_ES\_UNKNOWN, 40
  - LEP\_HS\_CRITICAL\_FAILURE, 42
  - LEP\_HS\_DEGRADED\_WARNING, 42
  - LEP\_HS\_MAJOR\_FAILURE, 42
  - LEP\_HS\_MINOR\_FAILURE, 42
  - LEP\_HS\_NON\_RECOVERABLE\_ERROR, 42
  - LEP\_HS\_OK, 42
  - LEP\_HS\_UNKNOWN, 42
  - LEP\_OPS\_ABORTED, 43



- LEP\_OPS\_COMPLETED, 43
- LEP\_OPS\_DEGRADED, 43
- LEP\_OPS\_DORMANT, 43
- LEP\_OPS\_ERROR, 43
- LEP\_OPS\_IN\_SERVICE, 43
- LEP\_OPS\_LOST\_COMMUNICATION, 43
- LEP\_OPS\_NO\_CONTACT, 43
- LEP\_OPS\_NON\_RECOVERABLE\_ERROR, 43
- LEP\_OPS\_OK, 43
- LEP\_OPS\_OTHER, 43
- LEP\_OPS\_POWER\_MODE, 43
- LEP\_OPS\_PREDICTIVE\_FAILURE, 43
- LEP\_OPS\_STARTING, 43
- LEP\_OPS\_STOPPED, 43
- LEP\_OPS\_STOPPING, 43
- LEP\_OPS\_STRESSED, 43
- LEP\_OPS\_SUPPORTING\_ENTITY\_IN\_ERROR, 43
- LEP\_OPS\_UNKNOWN, 43
- LEP\_OS\_ABORTED, 42
- LEP\_OS\_COMPLETED, 42
- LEP\_OS\_DORMANT, 42
- LEP\_OS\_EMIGRATING, 42
- LEP\_OS\_IMMIGRATING, 42
- LEP\_OS\_IN\_SERVICE, 42
- LEP\_OS\_IN\_TEST, 42
- LEP\_OS\_MIGRATING, 42
- LEP\_OS\_NOT\_AVAILABLE, 42
- LEP\_OS\_SERVICING, 42
- LEP\_OS\_SHUTTING\_DOWN, 42
- LEP\_OS\_SNAPSHOTTING, 42
- LEP\_OS\_STARTING, 42
- LEP\_OS\_STOPPED, 42
- LEP\_OS\_STOPPING, 42
- LEP\_OS\_TRANSITIONING, 42
- LEP\_OS\_UNKNOWN, 42
- LEP\_PS\_DEGRADED, 43
- LEP\_PS\_ERROR, 43
- LEP\_PS\_OK, 43
- LEP\_PS\_UNKNOWN, 43
- LEP\_RS\_DEFERRED, 41
- LEP\_RS\_DISABLED, 40
- LEP\_RS\_ENABLED, 40
- LEP\_RS\_NO\_CHANGE, 41
- LEP\_RS\_NOT\_APPLICABLE, 41
- LEP\_RS\_OFFLINE, 41
- LEP\_RS\_QUIESCE, 41
- LEP\_RS\_REBOOT, 41
- LEP\_RS\_RESET, 41
- LEP\_RS\_SHUT\_DOWN, 40
- LEP\_RS\_TEST, 41
- LEP\_RS\_UNKNOWN, 40
- OSBase\_NextHopIPRoute.h
- NH\_AT\_IPV4, 61
- NH\_AT\_IPV6, 61
- NH\_AT\_UNKNOWN, 61
- NH\_RTD\_BGP, 61
- NH\_RTD\_CONNECTED, 61
- NH\_RTD\_EGP, 61
- NH\_RTD\_EIGRP, 61
- NH\_RTD\_HELLO, 61
- NH\_RTD\_IGRP, 61
- NH\_RTD\_ISIS, 61
- NH\_RTD\_OSPF, 61
- NH\_RTD\_OTHER, 61
- NH\_RTD\_RIP, 61
- NH\_RTD\_UNKNOWN, 61
- NH\_RTD\_USER\_DEFINED, 61
- NH\_RTS\_HOST, 61
- NH\_RTS\_LINK, 61
- NH\_RTS\_NOWHERE, 61
- NH\_RTS\_SITE, 61
- NH\_RTS\_UNIVERSE, 61
- NH\_RTT\_COMPAT, 62
- NH\_RTT\_DEFAULT, 62
- NH\_RTT\_LOCAL, 62
- NH\_RTT\_MAIN, 62
- NH\_RTT\_UNSPEC, 62
- NH\_RTTY\_ANYCAST, 62
- NH\_RTTY\_BLACKHOLE, 62
- NH\_RTTY\_BROADCAST, 62
- NH\_RTTY\_LOCAL, 62
- NH\_RTTY\_MULTICAST, 62
- NH\_RTTY\_NAT, 62
- NH\_RTTY\_PROHIBIT, 62
- NH\_RTTY\_THROW, 62
- NH\_RTTY\_UNICAST, 62
- NH\_RTTY\_UNREACHABLE, 62
- NH\_RTTY\_UNSPEC, 62
- NH\_RTTY\_XRESOLVE, 62
- NH\_TOR\_ACTUAL, 62
- NH\_TOR\_ADMINISTRATOR, 62
- NH\_TOR\_COMPUTED, 62
- OSBase\_CSHostedRouteProviderAssociationCleanup  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderAssociatorNames  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderAssociators  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderCleanup  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderCreateInstance  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderDeleteInstance  
  cmpiOSBase\_CSHostedRouteProvider.c, 70
- OSBase\_CSHostedRouteProviderEnumInstanceNames  
  cmpiOSBase\_CSHostedRouteProvider.c, 70



- OSBase\_CSHostedRouteProviderEnumInstances  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_CSHostedRouteProviderExecQuery  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_CSHostedRouteProviderGetInstance  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_CSHostedRouteProviderReferenceNames  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_CSHostedRouteProviderReferences  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_CSHostedRouteProviderSetInstance  
  cmpiOSBase\_CSHostedRouteProvider.c, 71
- OSBase\_LANEndpoint.c  
  changeLinkOPState, 95  
  datetime\_str\_interval\_to\_ms, 95  
  freeLANEndpoint, 95  
  freeLANEndpointList, 96  
  getALLLANEndpoints, 96  
  getLANEndpoint, 96  
  getLANEndpoints, 96  
  nlInfoTOLanEP, 96  
  nlListTOLanEPList, 97
- OSBase\_LANEndpoint.h  
  changeLinkOPState, 44  
  CREATION\_CLASS\_NAME, 39  
  freeLANEndpoint, 44  
  freeLANEndpointList, 44  
  getALLLANEndpoint, 44  
  getLANEndpoint, 45  
  getLANEndpoints, 45  
  LANENDPOINT\_CAPTION, 39  
  LANENDPOINT\_DESC, 39  
  MAXHOSTNAMELEN, 39  
  nlInfoTOLanEP, 45  
  nlListTOLanEPList, 45
- OSBase\_LANEndpointProviderCleanup  
  cmpiOSBase\_LANEndpointProvider.c, 76
- OSBase\_LANEndpointProviderCreateInstance  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderDeleteInstance  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderEnumInstanceNames  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderEnumInstances  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderExecQuery  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderGetInstance  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderInvokeMethod  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderMethodCleanup  
  cmpiOSBase\_LANEndpointProvider.c, 77
- OSBase\_LANEndpointProviderSetInstance  
  cmpiOSBase\_LANEndpointProvider.c, 78
- OSBase\_Netlink.c  
  nlAddAttrToMsg, 99  
  nlAddAttrToMsg32, 99  
  nlAddLinkToList, 100  
  nlAddr\_n2a, 100  
  nlAddRouteToList, 100  
  nlCloseSocket, 101  
  nlCreateDefaultLinkInfo, 101  
  nlCreateDefaultRtInfo, 101  
  nlGenLinkFilter, 101  
  nlGenRouteFilter, 102  
  nlGetLinks, 102  
  nlGetLinkTypePos, 102  
  nlGetRoutes, 102  
  nlModifyLink, 103  
  nlModifyRoute, 103  
  nlOpenSocket, 103  
  nlResetLinkFilter, 103  
  nlResetRouteFilter, 104  
  nlSH, 104
- OSBase\_Netlink.h  
  address, 54  
  ARRAY\_SIZE, 49  
  broadcast, 54  
  change, 55  
  dstAddr, 55  
  dstLen, 55  
  family, 55  
  flags, 55  
  FREE\_SAFE, 49  
  gw, 55  
  ifname, 55  
  index, 55  
  inputIf, 55  
  link, 55  
  linkFlt, 55  
  linkmode, 56  
  linkType, 56  
  linkTypeName, 56  
  map, 56  
  metrics, 57  
  mtu, 57  
  NL\_SOCKET\_RCV\_BUFF\_LEN, 49  
  NL\_SOCKET\_SND\_BUFF\_LEN, 49  
  nlAddAttrToMsg, 50  
  nlAddAttrToMsg32, 50  
  nlAddLinkToList, 50  
  nlAddr\_n2a, 51  
  nlAddRouteToList, 51  
  nlCloseSocket, 51  
  nlCreateDefaultLinkInfo, 51  
  nlCreateDefaultRtInfo, 52  
  nlGenLinkFilter, 52

- nlGenRouteFilter, 52
- nlGetLinks, 52
- nlGetLinkTypePos, 53
- nlGetRoutes, 53
- nlModifyLink, 53
- nlModifyRoute, 54
- NLMSG\_TAIL, 49
- nlOpenSocket, 54
- nlResetLinkFilter, 54
- nlResetRouteFilter, 54
- operstate, 57
- outputIf, 57
- prefSrc, 57
- priority, 57
- protocol, 57
- qdisc, 57
- rtFt, 57
- scope, 57
- srcAddr, 57
- srcLen, 58
- stats, 58
- table, 58
- tos, 58
- txqlen, 58
- type, 58
- OSBase\_NextHopIPRoute.c
  - addIPRoute, 106
  - delIPRoute, 106
  - freeNextHopIP, 106
  - freeNextHopIPList, 106
  - getAddrType, 106
  - getAddrTypeStr, 107
  - getAllIPRoutes, 107
  - getIPRouteId, 107
  - getIPRoutes, 107
  - nhTONInfo, 108
  - nlInfoTONh, 108
  - nlListTONhList, 108
- OSBase\_NextHopIPRoute.h
  - addIPRoute, 63
  - delIPRoute, 63
  - freeNextHopIP, 63
  - freeNextHopIPList, 63
  - getAddrType, 63
  - getAddrTypeStr, 64
  - getAllIPRoutes, 64
  - getIPRouteId, 64
  - getIPRoutes, 64
  - INSTANCEID\_FORMAT, 60
  - INSTANCEID\_FORMAT\_PARSE, 60
  - nhTONInfo, 65
  - nlInfoTONh, 65
  - nlListTONhList, 65
- OSBase\_NextHopIPRouteProviderCleanup
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderCreateInstance
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderDeleteInstance
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderEnumInstanceNames
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderEnumInstances
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderExecQuery
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderGetInstance
  - cmpiOSBase\_NextHopIPRouteProvider.c, 83
- OSBase\_NextHopIPRouteProviderInvokeMethod
  - cmpiOSBase\_NextHopIPRouteProvider.c, 84
- OSBase\_NextHopIPRouteProviderMethodCleanup
  - cmpiOSBase\_NextHopIPRouteProvider.c, 84
- OSBase\_NextHopIPRouteProviderSetInstance
  - cmpiOSBase\_NextHopIPRouteProvider.c, 84
- OSBase\_RouteUsesEndpointProviderAssociationCleanup
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderAssociatorNames
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderAssociators
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderCleanup
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderCreateInstance
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderDeleteInstance
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderEnumInstanceNames
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 91
- OSBase\_RouteUsesEndpointProviderEnumInstances
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- OSBase\_RouteUsesEndpointProviderExecQuery
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- OSBase\_RouteUsesEndpointProviderGetInstance
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- OSBase\_RouteUsesEndpointProviderReferenceNames
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- OSBase\_RouteUsesEndpointProviderReferences

- cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- OSBase\_RouteUsesEndpointProviderSetInstance
  - cmpiOSBase\_RouteUsesEndpointProvider.c, 92
- otherDerivation
  - nextHopIP, 14
- otherEnabledState
  - LANEndpoint, 10
- otherTypeDescription
  - LANEndpoint, 10
- outputIf
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- prefixLength
  - nextHopIP, 14
- prefSrc
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- primaryStatus
  - LANEndpoint, 10
- priority
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- protocol
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- protocolIFType
  - LANEndpoint, 10
- qdisc
  - nlLinkInfo, 19
  - OSBase\_Netlink.h, 57
- requestedState
  - LANEndpoint, 11
- routeDerivation
  - nextHopIP, 14
- routeGateway
  - nextHopIP, 15
- routeMetric
  - nextHopIP, 15
- routeOutputIf
  - nextHopIP, 15
- routeScope
  - nextHopIP, 15
- routeTable
  - nextHopIP, 15
- routeType
  - nextHopIP, 15
- rtFlt
  - OSBase\_Netlink.h, 57
- scope
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- sptr
  - LANEndpointList, 12
  - nextHopIPList, 16
  - nlLinkInfoList, 20
  - nlRouteInfoList, 24
- srcAddr
  - nlRouteInfo, 22
  - OSBase\_Netlink.h, 57
- srcLen
  - nlRouteInfo, 23
  - OSBase\_Netlink.h, 58
- stats
  - nlLinkInfo, 19
  - OSBase\_Netlink.h, 58
- statusDescriptions
  - LANEndpoint, 11
- systemCreationClassName
  - LANEndpoint, 11
- systemName
  - LANEndpoint, 11
- table
  - nlRouteInfo, 23
  - OSBase\_Netlink.h, 58
- tos
  - nlRouteInfo, 23
  - OSBase\_Netlink.h, 58
- transitioningToState
  - LANEndpoint, 11
- txqlen
  - nlLinkInfo, 19
  - OSBase\_Netlink.h, 58
- type
  - nlLinkInfo, 19
  - nlRouteInfo, 23
  - OSBase\_Netlink.h, 58
- typeOfRoute
  - nextHopIP, 15