

NAME

mib – Load and query SNMP MIB definitions.

DESCRIPTION

The **mib** command allows to load and query definitions contained in SNMP Management Information Base (MIB) specifications (RFC 1155, RFC 1212, RFC 1902 and RFC 1903). Queries are usually formulated for a specific node in the global MIB tree. The **mib** command also allows to format/scan MIB values from/to their underlying primitive representation.

MIB NODE NAMES

A MIB node is either identified by the globally unique object identifier in dotted notation (e.g. 1.3.6.1.2.1.1.1) or by its descriptive name (e.g. sysDescr). However, simple names are only unique for a single MIB module. A globally unique name is therefore the combination of the MIB module name and the name of a MIB node defined in the module (e.g. SNMPv2-MIB!sysDescr). The ! character separates the optional MIB module name from the node name.

It is also legal to have hexadecimal sub-identifier in an object identifier. A colon instead of a dot is used to indicate that the following sub-identifier is a hexadecimal value. You can also use a dot followed by the 0x prefix. This is useful to encode character strings in an object identifier. As an example, the object identifier 1.3.6.1.4.1.0x627:74:75:62:73 will be accepted and converted to 1.3.6.1.4.1.1575.116.117.98.115.

It is suggested to use the object identifier notation whenever possible in scripts because it is globally unique and fast. Globally unique names (like SNMPv2-MIB!sysDescr) are also a good and fast choice. Simple names may lead to ambiguities and can result in slower lookups if the name can not be found in the underlying hash table. Composed simple names (like system.sysDescr) make ambiguities less likely but they may cause substantial overhead.

MIB COMMAND**mib load file**

The **mib load** command loads the MIB definitions contained in *file*. The file should contain a valid MIB definition. The built-in parser reads the *file*. Parsing errors are written to stderr. The MIB file is usually located at the file system position defined by the *file* argument. However, if *file* is a simple file name and it does not exist in the current directory, then it will be searched in \$tnm(library)/site and \$tnm(library)/mibs in this order. A condensed format of the MIB definition is saved in a platform specific sub-directory below \$tnm(library) to speed up future load commands. Note, this requires write permissions for the platform specific sub-directory.

The Tnm extension automatically loads all MIB files named in the global Tcl variable \$tnm(mibs). This allows to pre-define a set of useful MIB definitions. The default is the set of IETF MIBs. It is possible to bypass the auto-loading mechanism if the first **mib** command is a **mib load** command. Note that the **snmp** command may also invoke **mib** commands.

mib [-exact] name label

The **mib name** command returns the short name associated with a node in the MIB tree. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. For example, **mib name 1.3.6.1.2.1.1.3.0** will return sysUpTime.0 while **mib -exact name 1.3.6.1.2.1.1.3.0** will fail.

mib [-exact] *oid label*

The **mib oid** command is the counterpart to the **mib name** command and returns the object identifier of the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. If a *label* is not unique within the loaded MIB definitions, the object identifier of the first matching MIB node will be returned.

mib [-exact] **module** *label*

The **mib module** command returns the name of the module which defines the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib [-exact] **macro** *label*

The **mib macro** command returns the name of the ASN.1 macro which has been used to define the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib [-exact] **syntax** *label*

The **mib syntax** command returns the ASN.1 syntax associated with a MIB node or an empty string if the node identified by *label* has no associated ASN.1 syntax. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. The syntax name returned is the syntax as defined in the macro. This might be the name of a textual-convention. The **mib tc** command can be used to get the underlying base syntax which is used to encode values on the network.

mib [-exact] **description** *label*

The **mib description** command returns the textual description of the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib [-exact] **successor** *label*

The **mib successor** command returns a list of all known successors of the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. The format of the result depends on the format of *label*: If *label* is an object identifier, then the result will be a list of object identifier. The result is a list of names if *label* is a name of a MIB node.

mib [-exact] **parent** *label*

The **mib parent** command returns the parent node of the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. The format of the result depends on the format of *label*: If *label* is an object identifier, then the result is the object identifier of the parent node. The result is the name of the parent node if *label* is a name of a MIB node.

mib [-exact] **access** *label*

The **mib access** command returns the max-access for a MIB node or an empty string if there is no access associated with the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier. The returned access mode is one of the values not-accessible,

read-only, read-create, read-write, write-only.

mib [-exact] **index** *label*

The **mib index** command returns the list of MIB names which are used as a key in a conceptual table or an empty result if the MIB node identified by *label* does not correspond to a MIB table. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib [-exact] **tc** *label*

The **mib tc** command returns a Tcl list which describes the textual convention associated with the MIB node identified by *label*. The result is an empty string if the MIB node does not make use of a textual convention. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

The list contains the following elements: the name of the textual convention, the name of the underlying ASN.1 type, the format string (display hint), a list of name value pairs used to convert enumerations to integer values, the MIB module which contains the definition of the textual convention and the file which contains the definition.

mib [-exact] **file** *label*

The **mib file** command returns the file name that contains the definition of the MIB node identified by *label*. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib format *label value*

The **mib format** command can be used to apply a textual convention that is defined for the MIB node identified by *label* to *value*. This command is used internally to return readable values for SNMP request and may be helpful in situations where it is appropriate to apply these formatting rules directly. The **mib format** command can be used to render integer enumerations, to apply display hints, to render time tick values and to convert object identifier into globally unique names. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib scan *label value*

The **mib scan** command implements the inverse operation for the **mib format** command described above. It converts a rendered value into the underlying base string representation. This command may be useful in situations where you want to operate on the primitive format rather than the readable representation. The **mib scan** command returns the *value* unaltered if no conversions apply. The *label* may be a node name in one of the formats discussed above. The -exact option forces strict lookups where it is not allowed that *label* contains an instance identifier.

mib walk *varName label body*

The **mib walk** command evaluates the Tcl script *body* for each known MIB node below the node identified by *label*. The **mib walk** command allows to walk entire MIB subtrees. Every nodes in a subtree identified by *label* is assigned to *varName* before the *body* is evaluated. The value assigned to *varName* depends on the format of *label*: If *label* is an object identifier, then the value will be an object identifier. The value of *varName* is the name of a MIB node if *label* is a name of a MIB node.

SEE ALSO

scotty(1), Tnm(n), Tcl(n)

AUTHORS

Juergen Schoenwaelder <schoenw@cs.utwente.nl>

Sven Schmidt <vschmidt@ibr.cs.tu-bs.de>

Erik Schoenfelder <schoenfr@gaertner.de>