SilverCreek SNMP Libraries and APIs

What's Inside:

Replace the "Scotty" Libraries

The SilverCreek SNMP APIs replace the Scotty libraries for software developers seeking a migration path from the legacy Scotty libraries to SilverCreek libraries that support SNMPv3 security. The SilverCreek SNMP libraries replace the Scotty libraries with a complete set of tools, full support, developer's guides, updates, and even more. The SilverCreek libraries represent the ideal solution for creating network management applications and/or migrating those applications from the Scotty libraries.

SilverCreek's APIs are written in C++ and are accessible through a Tcl shell or command line or a Tcl program. Tcl, the Tool Command Language, is a flexible, interpretive language, popular for network applications.

APIs are application programming interfaces that execute requests for services (for example getting the value of an SNMP object identifier) on behalf of a program making the request.

Reduce Development Time

Good quality and mature APIs can greatly reduce development time, because the developer can use the APIs as building blocks to create his application, without having to create the building blocks themselves.

With SilverCreek SL, developers can migrate home grown applications from obsolete and/or unsupported libraries and APIs, thereby preserving the investment in these applications.

Full Support for SNMPv3

Scotty/Tnm, a freely available, open source, set of SNMP APIs is a popular foundation for many applications. Scotty's authors, however, stopped work on Scotty in 2002 and moved on to other projects. As a result, applications based on the Scotty APIs do not have support for the important security features of SNMPv3.

Fortunately, the SilverCreek SL SNMP APIs, provide full support for SNMPv3 security.

A Scotty SNMP API to SilverCreek SL SNMP API mapping is included below.

Comprehensive Tools and Documentation

SilverCreek SL SNMP APIs also include a comprehensive set of tools and documentation:

- SNMP Command Tool — Perform SNMP operations on Object Names or OIDs
- SNMP Command Wizard — Perform SNMP requests using multiple objects or variable bindings
- SNMPv3 USM Manager — Create and manage SNMPv3 users, groups, and MIB views
- Polling Tool — Poll the SNMP agent for any number of objects at a user specified interval
- Performance Monitoring Tool — Measure the response time to certain operations with a device and compare to another device
- Notification Configuration Tool — Create and manage SNMP notification and filter tables
- Notification Monitor Tool — Visually examine and automatically test the contents of all traps, alerts or notifications sent over port 162
- MIB Table Inspector — View, update, and insert rows in a selected MIB Table
- MIB Compiler — Compiles private or IETF standard MIBs
- MIB Browser — A Multi-purpose tool that is a MIB walker (displaying actual MIBs implemented in your agent, a MIB browser (displaying all locally loaded MIBs), and a MIB Info lookup capability to get the details on a particular MIB object

The API Reference and Developer Guide is included with the SilverCreek distribution and may be found under the directory Samples/CLI/

SilverCreek to Scotty SNMP API Mapping

The SilverCreek SNMP APIs replace the Scotty libraries for software developers seeking a migration path from the legacy Scotty libraries to SilverCreek libraries that support SNMPv3 security. The SilverCreek SNMP libraries replace the Scotty libraries with a complete set of tools, full support, developer's guides, updates, and even more. The SilverCreek libraries represent the ideal solution for creating network management applications and/or migrating those applications from the Scotty libraries.

This is an overview for those who wish to migrate from Scotty/Tnm to the SilverCreek API.

Part A: Mapping the Scotty 'mib' command for querying MIB module properties.

Part B: Mapping the Scotty 'snmp' Command for sending SNMP requests.

Note: Scotty 3.0 version moves the "mib" and "snmp" commands into the Tnm namespace from the global namespace in 2.0. Otherwise their usages are similar.

Some Scotty/Tnm commands do not have an equivalent in SilverCreek; often these commands are not core SNMP/MIB commands and are not in the "must-have" category. To request that a Scotty/Tnm command be added to the SilverCreek API, please send the request to IWL Support.

Many SilverCreek API commands for SNMPv3 context configuration, passwordToKey, keyChange, etc. are provided with the SilverCreek API, but not in Scotty/Tnm.

Part A: Migrating the Scotty 'mib' Command

Following is an overview of the Scotty MIB database related commands to SilverCreek API for querying MIB module properties. Typically in SilverCreek you use [snmpinfo mibprop object property]. However, if there are Agent Capabilities MIBs enabled you should use [<ctxproc> agentprop object property] that returns the syntax or access variation defined in the agent-cap statements instead.

Querying the ACCESS Property of a MIB object

Scotty: mib access node
  % mib access sysDescr

SilverCreek: snmpinfo mibprop object access
  % snmpinfo mibprop sysDescr access

Getting children of a MIB object

Scotty: mib children node
  % mib children system

SilverCreek: snmpinfo mibprop OID access
  % snmpinfo mibprop 1.3.6.1.2.1 immchildren

Querying the DISPLAY-HINT

Scotty: mib displayhint node
  % mib displayhint DisplayString

SilverCreek: snmpinfo mibprop object hint
  %snmpinfo mibprop sysDescr hint

Formatting a value according to a defined DISPLAY-HINT

Scotty: mib format nodename value
% mib format docsDevDateTime 00:00:01:01:00:00:00:00
>>0-1-1,0:0:0.0

SilverCreek: snmpinfo format object hint
%snmpinfo format docsDevDateTime 0x00:00:01:01:00:00:00:00
>> 0-1-1,0:0:0.0

Querying the MIB File name the defines the object

Scotty: mib file node
% mib file system

SilverCreek: snmpinfo mibprop object modulename snmpinfo mibprop modulename file
% snmpinfo mibprop sysDescr modulename
>> SNMPv2-MIB
% snmpinfo mibprop SNMPv2-MIB file
>>/home/xiangli/temp/mibs/rfc3418-SNMPv2-MIB.defs

In SilverCreek, the MIB file name may not be the same as the actual file name.

Querying the index objects of a columnar object

Scotty: mib index node
% mib index ifType index

SilverCreek: snmpinfo mibprop object index
%snmpinfo mibprop ifType index
>>IF-MIB:ifIndex

Note in SilverCreek the first argument 'object' must be an accessible columnar object in a table, otherwise this command will return an empty string. For example [snmpinfo mibprop ifTable index] will ONLY return an empty string.

Mapping OID to node name and vice versa

Scotty: mib name oid
mib oid name
% mib name 1.3.6.1.2.1.1.1
% mib oid sysDescr

SilverCreek: snmpinfo mibprop object value
snmpinfo mibprop oid descriptor

%snmpinfo mibprop 1.3.6.1.2.1.1.1 descriptor

%snmpinfo mibprop sysDescr value

Querying node type

Scotty: mib macro node

% mib macro sysDescr

>> OBJECT-TYPE

SilverCreek: snmpinfo mibprop object type

%snmpinfo mibprop sysDescr type

>> OBJECT-TYPE

Querying Group Members

Scotty: mib member node

% Tnm::mib member systemGroup

>> sysDescr sysObjectID sysUpTime sysContact ...

SilverCreek: snmpinfo groupname objects

% snmpinfo systemGroup objects

>> sysDescr sysObjectID sysUpTime sysContact ...

Querying a node's 'next' node in lexicographically order

Scotty: Not supported

SilverCreek: snmpinfo mibprop object next

% snmpinfo mibprop ifType next

>>IF-MIB:ifMtu

Querying a node's 'parent' node

Scotty: mib parent node

SilverCreek: snmpinfo mibprop object parent

% snmpinfo mibprop ifType parent

>>IF-MIB:ifEntry

Querying the Size/Range

Scotty: mib range type / mib size type

% mib size DisplayString

>> 0 255

SilverCreek: snmpinfo mibprop object range snmpinfo mibprop object size

Querying and mapping the Enumerations

Scotty: mib enums type

% mib enums IANAifType

SilverCreek: snmpinfo mibprop object enums numToEnum object number enumToNum object label

%snmpinfo mibprop ifType enums

%numToEnum ifType 6

>> ethernetCsmacd

%enumToNum ifType ethernetCsmacd

>>6

Querying the STATUS

Scotty: mib status node

% mib status sysDescr

SilverCreek: snmpinfo mibprop object status

%snmpinfo mibprop sysDescr status

>> current

Querying the SYNTAX and base syntax of Textual Convention

Scotty: mib type node -- Syntax mib syntax node --base syntax

% mib syntax sysDescr

>> OCTET STRING

% mib type sysDescr

>>SNMPv2-TC!DisplayString

SilverCreek: snmpinfo mibprop object syntax snmpinfo mibprop object base-syntax snmpinfo mibprop sysDescr syntax

>>SNMPv2-TC:DisplayString

snmpinfo mibprop sysDescr basesyntax

>>OctetString

Querying details about name and index of an OID

Scotty: mib unpack oid mib split oid

% mib unpack sysORDescr.6

>> 6

% mib split sysORDescr.6

>> 1.3.6.1.2.1.1.9.1.3 6

SilverCreek: snmpinfo mibprop OID full

This command returns:

object-name | object type | object identifier | index part| parsed index

For example:

% set result [snmpinfo mibprop 1.3.6.1.2.1.7.5.1.1.0.0.0.0.1003 full]

>>UDP-MIB:udpLocalAddress.0.0.0.0.1003 OBJECT-TYPE 1.3.6.1.2.1.7.5.1.1 0.0.0.0.1003 {0.0.0.0 1003}

object-name | object type | object identifier | index part | parsed index)

To get the name: %lindex $result 0

To get the object type: %lindex $result 1

To get the object ID: %lindex $result 2

To get the index part: %lindex $result 3

%snmpinfo mibprop 1.3.6.1.2.1.7.5.1.1.0.0.0.0.1003 index

>> UDP-MIB:udpLocalAddress UDP-MIB:udpLocalPort

So we could get the value for each index object from the 'parsed index' part in the results of the command [snmpinfo mibprop oid full]

To get the value for the first index object udpLocalAddress:

%lindex [lindex $result 4 0]

To get the value for the second index object udpLocalPort :

%lindex [lindex $result 4 1]

Querying objects that should be included in a notification

Scotty: mib variables node

% mib variables linkUp

>> ifIndex ifAdminStatus ifOperStatus

SilverCreek: snmpinfo mibprop object varlist

% snmpinfo mibprop linkUp varlist

>> ifIndex ifAdminStatus ifOperStatus

Retrieving names of all loaded objects

Scotty: Not Supported

SilverCreek: snmpinfo objects -nomodule snmpinfo objects -withmodule

Retrieving a list of all loaded MIBs

Scotty: Not Supported

SilverCreek: snmpinfo mibprop modules

Constructing object identifier values for columnar in a table.

Scotty: mib pack oid value1 value2 … valuen

SilverCreek snmptcl::oidpack object/oid value1 value2 … valuen

The oid/object argument identifies a conceptual row. The values are used to build an instance identifier according to the object's index components. The command returns the complete object identifier needed to access the variable.

%snmptcl::oidpack usmUserStatus 0x12:34:56:67:89 md5desuser

>>1.3.6.1.6.3.15.1.2.2.1.13.5.18.52.86.103.137.10.109.100.53.100.101.1
15.117.11

5.101.114

%snmptcl::oidpack snmpCommunityName BB

>>1.3.6.1.6.3.18.1.1.1.2.66.66

%snmptcl::oidpack atIfIndex 192.168.1.0

>>1.3.6.1.2.1.3.1.1.1.1.192.168.1.0

%snmptcl::oidpack udpLocalPort 192.168.1.0 1900

>>1.3.6.1.2.1.7.5.1.2.192.168.192.1.1900

%snmptcl::oidpack vacmViewTreeFamilyMask ApplicationsView
1.3.6.1.6.3.16.1.5.2

>>1.3.6.1.6.3.16.1.5.2.1.3.16.65.112.112.108.105.99.97.116.105.111.11
0.115.86.1

05.101.119.10.1.3.6.1.6.3.16.1.5.2

%snmptcl::oidpack snmpNotifyFilterType wellKnownTraps
1.3.6.1.6.3.1.1.5

>>1.3.6.1.6.3.13.1.3.1.3.14.119.101.108.108.75.110.111.119.110.84.114
.97.112.11

5.1.3.6.1.6.3.1.1.5

Part B: Migrating the Scotty 'snmp' Command

Following is an overview of the Scotty SNMP Session related commands to the SilverCreek API for sending SNMP requests. Typically in SilverCreek you use commands in 'snmptcl' namespace.

Creating SNMP generator Sessions

Scotty: snmp session [option value ...]

% set session1 [snmp session -address 192.168.123.163 -version SNMPv1]

SilverCreek: snmptcl::context::create [option value...]

% set session1 [snmptcl:context::create -address 192.168.123.163 -version SNMPv1]

Sending SNMP GET Requests

Scotty: session get varBindList

% $session1 get SNMPv2-MIB!sysDescr.0 SNMPv2-MIB!sysName.0

SilverCreek: snmptcl::snmpget $args

Examples:

% snmptcl::snmpget SNMPv2-MIB:sysDescr.0 \

SNMPv2-MIB:sysName.0 -context $session1

% snmptcl::snmpget -vbinds {sysDescr.0 sysName.0} \

-rvbinds result -context $session1

'args' can contain various option/value pairs.

For details please contact IWL Support for the SilverCreek Developer's Guide.

Sending SNMP GET-NEXT Requests

Scotty: $session getnext varBindList

% $session1 getnext {{sysDescr} {sysContact}}

SilverCreek: snmptcl::snmpnext args

%set startoid1 sysDescr

%set startoid2 sysContact

%snmptcl::snmpnext startoid1 value1 startoid2 value2 -context $session1

'args' can contain various option/value pairs.

For details please contact IWL Support for the SilverCreek Developer's Guide.

Sending SNMP BULK Requests

Scotty: $session getbulk nr mr vbl

% $session1 getbulk sysDescr 1 10 {{sysDescr.0} {sysContact}}

SilverCreek: snmptcl::snmpbulk args

% snmptcl::snmpbulk -nonrep 1 -maxrep 10 -vbinds {sysDescr sysContact} \

-rvbinds resultvar -context $session1

'args' can contain various option/value pairs.

For details please contact IWL Support for the SilverCreek Developer's Guide.

Sending SNMP SET Requests

Scotty: $session set varBindList

% $session1 set [list sysContact.0 "Xiang"]

SilverCreek: snmptcl::snmpset args

Examples:

% snmptcl::snmpset sysContact.0 "Xiang" -context $session1

'args' can contain various option/value pairs.

For details please contact IWL Support for the SilverCreek Developer's Guide.

Note Object's string instance can be specified as 'string' in the request.

!See also: Constructing object identifier values for columnar in a table.

snmptcl::oidpack command.

For example:

%snmptcl::snmpset object.4.abcd value1 -context $session1

%snmptcl::snmpset object.4.0x61:62:63:64 value1 -context $session1

%snmptcl::snmpset object.4.97.98.99.100 value1 -context $session1

Use SET request to create a row in a table:

Example 1:

%set index1 "1.98"

%set index2 "2.65.66"

%snmptcl::snmpset

pingCtlTargetAddressType.$index1.$index2 1

pingCtlTargetAddress.$index1.$index2 0x0a:d3:20:06

pingCtlAdminStatus.$index1.$index2 2

pingCtlRowStatus.$index1.$index2 4

Exampe 2:

%set index1 "1.98"

%set index2 "2.65.66"

%set vbl1 [list pingCtlTargetAddressType.$index1.$index2 \

InetAddressType 1]

%set vbl2 [list pingCtlTargetAddress.$index1.$index2 \

InetAddress $targetIp]

%set vbl3 [list pingCtlAdminStatus.$index1.$index2 INTEGER 2]

%set vbl4 [list pingCtlRowStatus.$index1.$index2 RowStatus 4]

%set stuff [snmptcl::snmpset -vbinds [list $vbl1 $vbl2 $vbl3 $vbl4]\

-context $Ctx -restatus eStatus -reindex eIndex]

Walking a MIB Branch (Subtree)

Scotty: $session walk varName varBindList

%$session1 walk res sysORTable

SilverCreek: snmptcl::snmpgetblock node value args

%snmptcl::snmpgetblock ifTable res -context $session1 -rindices rindices

Note The option -rindices let us get all 'index values' at the same time as retrieving table contents. This can be very useful in your scripts.

Scotty does not support this feature.

Destroying Sessions

Scotty: $session destroy

% set ses1 [snmp session -address 192.168.123.163 -version SNMPv1]

% $ses1 destroy

SilverCreek: $session destroy

% set ses1 [snmptcl:context::create -address 192.168.123.163 -version SNMPv1]

% $ses1 destroy

Receiving SNMP Notifications

Scotty: $session bind {} trap script

$session bind {} inform script

SilverCreek: $session register –add trap|inform callbackFunc

proc ::my_trap_in trapInfo {

puts “trap callback”

}

% set trapsession [snmptcl::context::create -traptransport IPv4 -trapport 162]

% $trapsession register -add trap ::my_trap_in

For details please see samples/CLI/snmp_trap_listener.tcl under your SilverCreek root directory.

(831) 460-7010
info@iwl.com