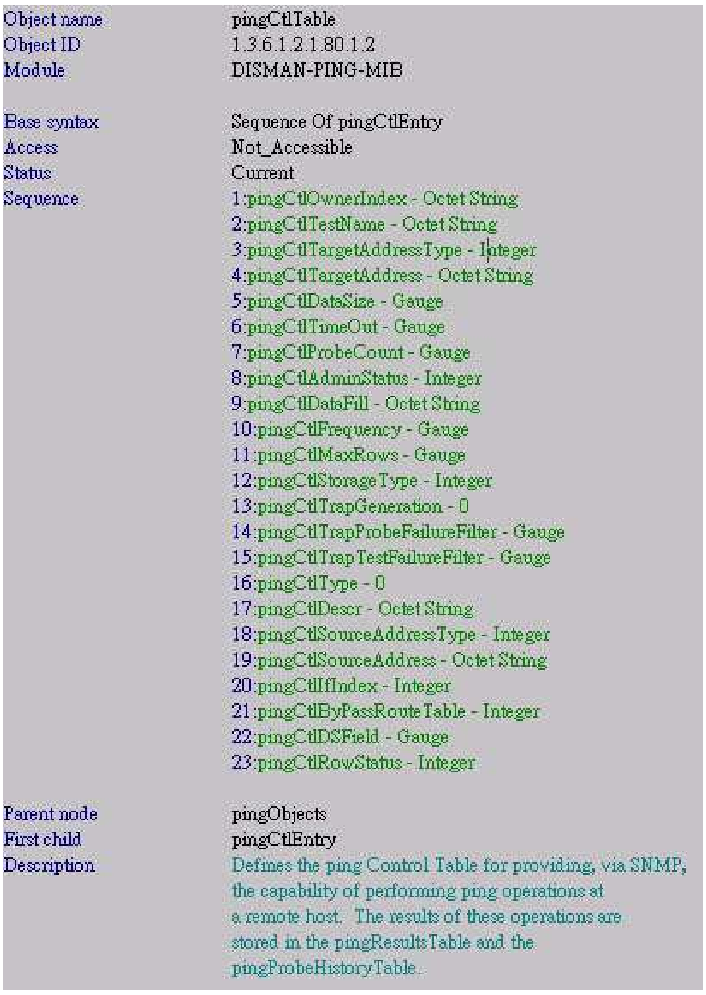
How to configure DISMAN-PING (RFC 2925P) via NETSNMP in DGS-3620 FW 1.02.R026?

The OIDs are defined in "**pingCtlTable**" of "**DISMAN-PING-MIB.mib**".

pingCtlTable (1.3.6.1.2.1.80.1.2)

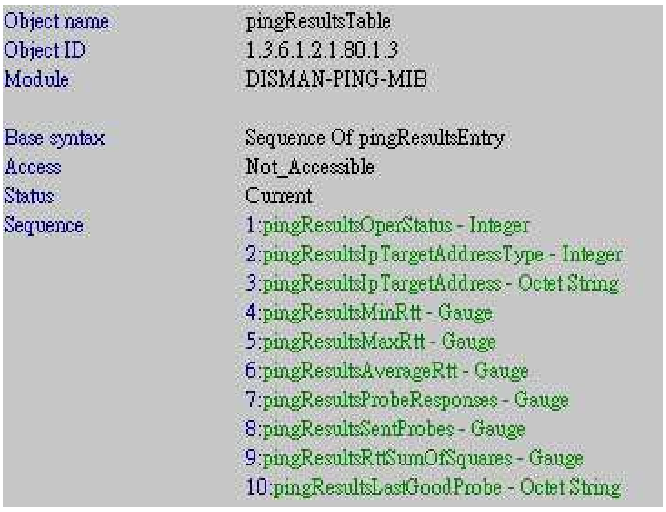
⇨ Setup the ping action.



Object name	pingCtlTable
Object ID	1.3.6.1.2.1.80.1.2
Module	DISMAN-PING-MIB
Base syntax	Sequence Of pingCtlEntry
Access	Not_Accessible
Status	Current
Sequence	1:pingCtlOwnerIndex - Octet String 2:pingCtlTestName - Octet String 3:pingCtlTargetAddressType - Integer 4:pingCtlTargetAddress - Octet String 5:pingCtlDataSize - Gauge 6:pingCtlTimeOut - Gauge 7:pingCtlProbeCount - Gauge 8:pingCtlAdminStatus - Integer 9:pingCtlDataFill - Octet String 10:pingCtlFrequency - Gauge 11:pingCtlMaxRows - Gauge 12:pingCtlStorageType - Integer 13:pingCtlTrapGeneration - 0 14:pingCtlTrapProbeFailureFilter - Gauge 15:pingCtlTrapTestFailureFilter - Gauge 16:pingCtlType - 0 17:pingCtlDescr - Octet String 18:pingCtlSourceAddressType - Integer 19:pingCtlSourceAddress - Octet String 20:pingCtlIfIndex - Integer 21:pingCtlByPassRouteTable - Integer 22:pingCtlDSField - Gauge 23:pingCtlRowStatus - Integer
Parent node	pingObjects
First child	pingCtlEntry
Description	Defines the ping Control Table for providing, via SNMP, the capability of performing ping operations at a remote host. The results of these operations are stored in the pingResultsTable and the pingProbeHistoryTable.

pingResultsTable (1.3.6.1.2.1.80.1.3)

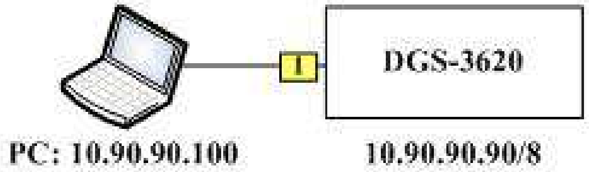
⇨ Check the ping status and result.



Object name	pingResultsTable
Object ID	1.3.6.1.2.1.80.1.3
Module	DISMAN-PING-MIB
Base syntax	Sequence Of pingResultsEntry
Access	Not_Accessible
Status	Current
Sequence	1:pingResultsOperStatus - Integer 2:pingResultsIpTargetAddressType - Integer 3:pingResultsIpTargetAddress - Octet String 4:pingResultsMinRtt - Gauge 5:pingResultsMaxRtt - Gauge 6:pingResultsAverageRtt - Gauge 7:pingResultsProbeResponses - Gauge 8:pingResultsSentProbes - Gauge 9:pingResultsRttSumOfSquares - Gauge 10:pingResultsLastGoodProbe - Octet String

For example

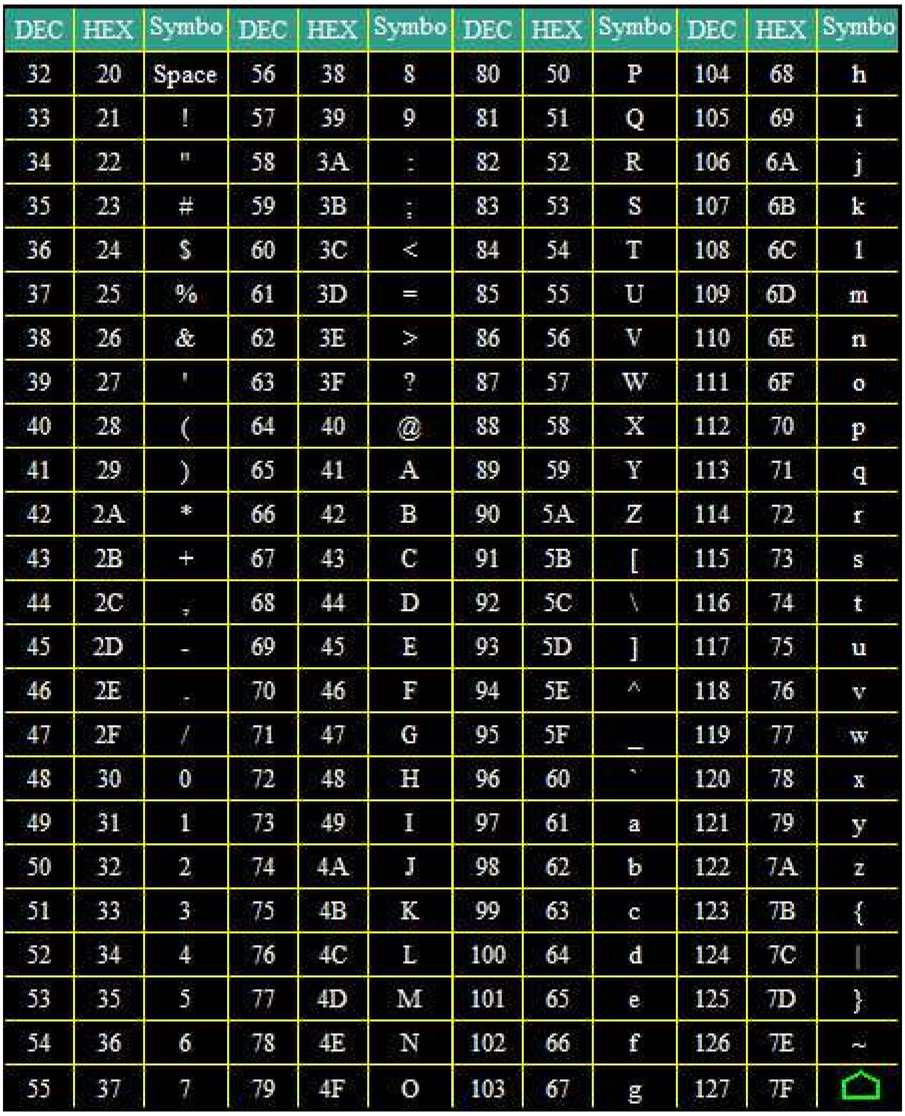
⇨ **DGS-3620** wants to ping the PC via SNMP as the topology below:



Step 1

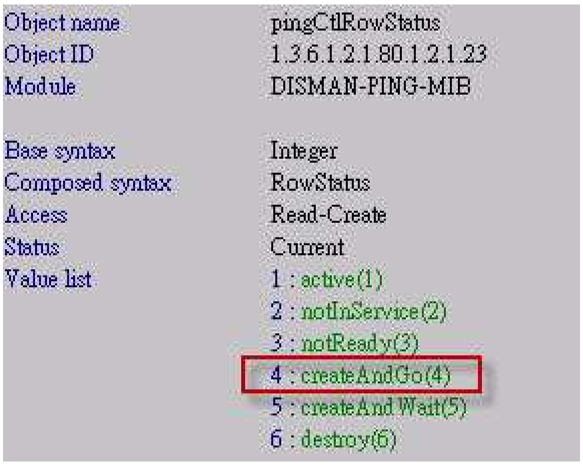
⇨ **Create a PING entry**

We should create an entry name: "**test**" for **pingCtlTable** first. After looking up ASCII Table below, the name "**test**" converts into "**116 101 115 116**"



DEC	HEX	Symbo	DEC	HEX	Symbo	DEC	HEX	Symbo	DEC	HEX	Symbo
32	20	Space	56	38	8	80	50	P	104	68	h
33	21	!	57	39	9	81	51	Q	105	69	i
34	22	"	58	3A	:	82	52	R	106	6A	j
35	23	#	59	3B	;	83	53	S	107	6B	k
36	24	$	60	3C	<	84	54	T	108	6C	l
37	25	%	61	3D	=	85	55	U	109	6D	m
38	26	&	62	3E	>	86	56	V	110	6E	n
39	27	'	63	3F	?	87	57	W	111	6F	o
40	28	(	64	40	@	88	58	X	112	70	p
41	29	)	65	41	A	89	59	Y	113	71	q
42	2A	*	66	42	B	90	5A	Z	114	72	r
43	2B	+	67	43	C	91	5B	[	115	73	s
44	2C	,	68	44	D	92	5C	\	116	74	t
45	2D	-	69	45	E	93	5D	]	117	75	u
46	2E	.	70	46	F	94	5E	^	118	76	v
47	2F	/	71	47	G	95	5F	_	119	77	w
48	30	0	72	48	H	96	60	`	120	78	x
49	31	1	73	49	I	97	61	a	121	79	y
50	32	2	74	4A	J	98	62	b	122	7A	z
51	33	3	75	4B	K	99	63	c	123	7B	{
52	34	4	76	4C	L	100	64	d	124	7C	|
53	35	5	77	4D	M	101	65	e	125	7D	}
54	36	6	78	4E	N	102	66	f	126	7E	~
55	37	7	79	4F	O	103	67	g	127	7F	⌂

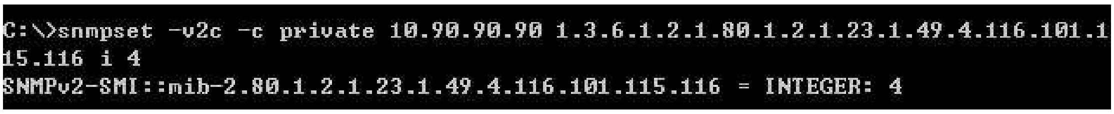
Then to create an entry via the MIB **pingCtlRowStatus** (1.3.6.1.2.1.80.1.2.1.23)



Object name	pingCtlRowStatus
Object ID	1.3.6.1.2.1.80.1.2.1.23
Module	DISMAN-PING-MIB
Base syntax	Integer
Composed syntax	RowStatus
Access	Read-Create
Status	Current
Value list	1 : active(1) 2 : notInService(2) 3 : notReady(3) 4 : createAndGo(4) 5 : createAndWait(5) 6 : destroy(6)

We should assign a index number "1"(ASCII = **49**) to this entry "**test**" (ASCII = **116.101.115.116**), and we have know "test" including "**4**" characters. So the result is:

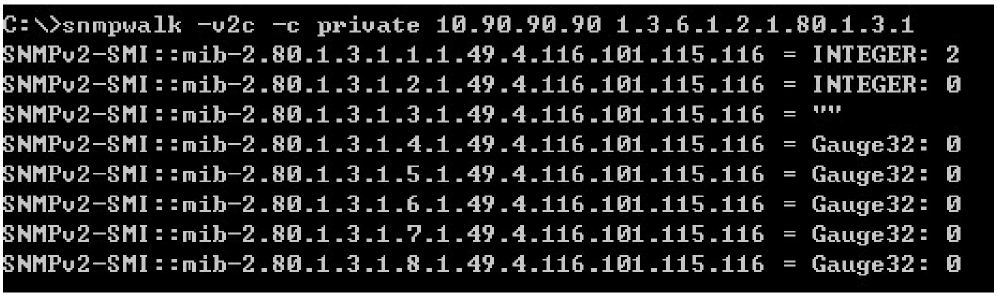
snmpset -v2c -c private 10.90.90.90 ***1.3.6.1.2.1.80.1.2.1.23.1.49.4.116.101.115.116 i 4***



```
C:\>snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.23.1.49.4.116.101.1
15.116 i 4
SNMPv2-SMI::mib-2.80.1.2.1.23.1.49.4.116.101.115.116 = INTEGER: 4
```

To check if it is successful by the command:

snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1

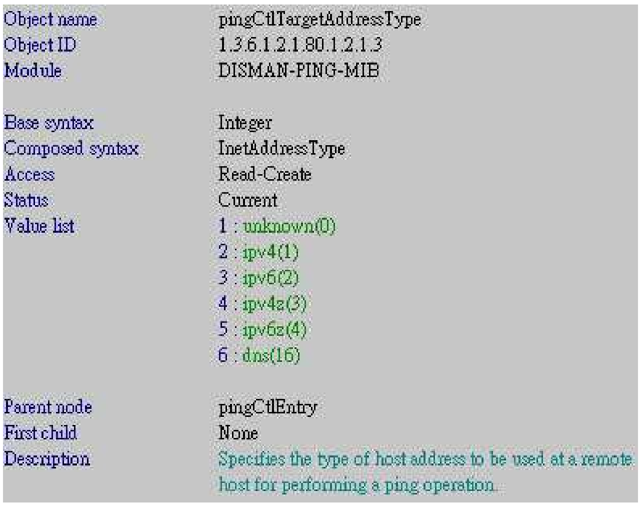


```
C:\>snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1
SNMPv2-SMI::mib-2.80.1.3.1.1.1.49.4.116.101.115.116 = INTEGER: 2
SNMPv2-SMI::mib-2.80.1.3.1.2.1.49.4.116.101.115.116 = INTEGER: 0
SNMPv2-SMI::mib-2.80.1.3.1.3.1.49.4.116.101.115.116 = ""
SNMPv2-SMI::mib-2.80.1.3.1.4.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.5.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.6.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.7.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.8.1.49.4.116.101.115.116 = Gauge32: 0
```

Step 2

⇨ **Specify a Target IP Type**

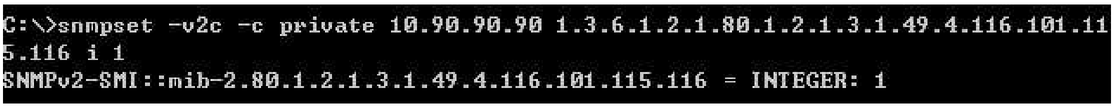
pingCtlTargetAddressType (1.3.6.1.2.1.80.1.2.1.3)



Object name	pingCtlTargetAddressType
Object ID	1.3.6.1.2.1.80.1.2.1.3
Module	DISMAN-PING-MIB
Base syntax	Integer
Composed syntax	InetAddressType
Access	Read-Create
Status	Current
Value list	1 : unknown(0) 2 : ipv4(1) 3 : ipv6(2) 4 : ipv4z(3) 5 : ipv6z(4) 6 : dns(16)
Parent node	pingCtlEntry
First child	None
Description	Specifies the type of host address to be used at a remote host for performing a ping operation.

Since PC’s IP is 10.90.90.100, we should set the target IP type to "IPv4", the command is:

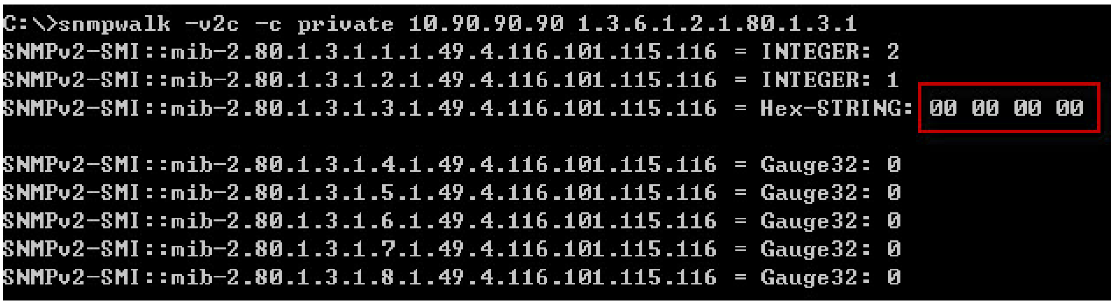
snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.3.1.49.4.116.101.115.116 ***i 1***



```
C:\>snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.3.1.49.4.116.101.11
5.116 i 1
SNMPv2-SMI::mib-2.80.1.2.1.3.1.49.4.116.101.115.116 = INTEGER: 1
```

To check if it is successful by the command:

snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1

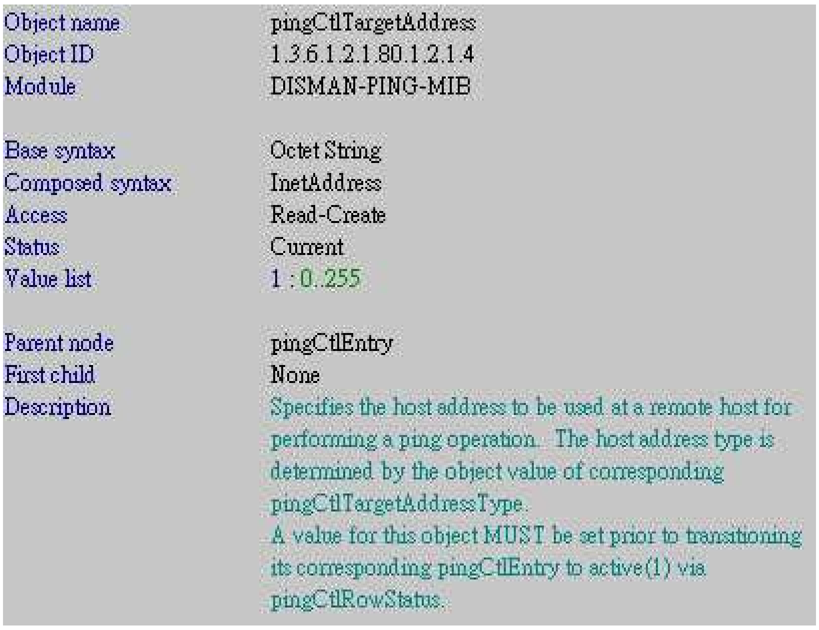


```
C:\>snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1
SNMPv2-SMI::mib-2.80.1.3.1.1.1.49.4.116.101.115.116 = INTEGER: 2
SNMPv2-SMI::mib-2.80.1.3.1.2.1.49.4.116.101.115.116 = INTEGER: 1
SNMPv2-SMI::mib-2.80.1.3.1.3.1.49.4.116.101.115.116 = Hex-STRING: 00 00 00 00

SNMPv2-SMI::mib-2.80.1.3.1.4.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.5.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.6.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.7.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.8.1.49.4.116.101.115.116 = Gauge32: 0
```

Step 3

⇨ **Specify a Target IP Address**

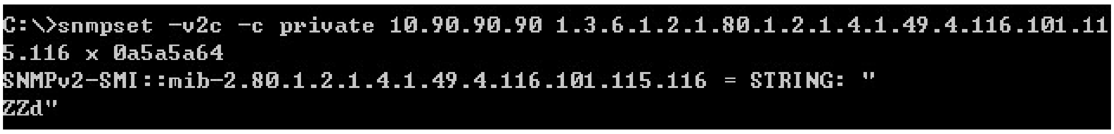
pingCtlTargetAddress (1.3.6.1.2.1.80.1.2.1.4)



Object name	pingCtlTargetAddress
Object ID	1.3.6.1.2.1.80.1.2.1.4
Module	DISMAN-PING-MIB
Base syntax	Octet String
Composed syntax	InetAddress
Access	Read-Create
Status	Current
Value list	1 : 0..255
Parent node	pingCtlEntry
First child	None
Description	Specifies the host address to be used at a remote host for performing a ping operation. The host address type is determined by the object value of corresponding pingCtlTargetAddressType. A value for this object MUST be set prior to transitioning its corresponding pingCtlEntry to active(1) via pingCtlRowStatus.

Firstly, we should convert the decimal IP address (10.90.90.100) into the hexadecimal IP address (0x0a.0x5a.0x5a.0x64), and then configure the IP address as the command:

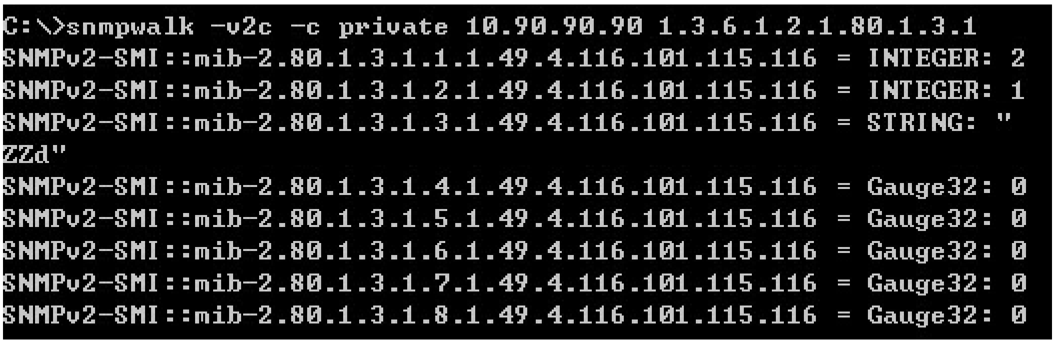
snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.4.1.49.4.116.101.115.116 x ***0a5a5a64***



```
C:\>snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.4.1.49.4.116.101.11
5.116 x 0a5a5a64
SNMPv2-SMI::mib-2.80.1.2.1.4.1.49.4.116.101.115.116 = STRING: "
ZZd"
```

To check if it is successful by the command:

snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1

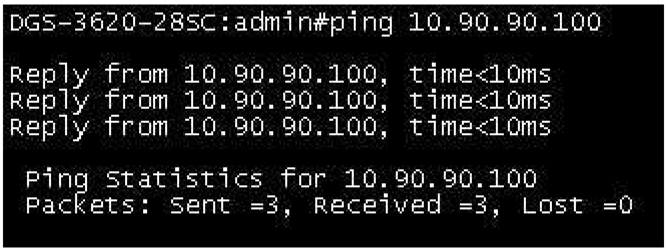


```
C:\>snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1
SNMPv2-SMI::mib-2.80.1.3.1.1.1.49.4.116.101.115.116 = INTEGER: 2
SNMPv2-SMI::mib-2.80.1.3.1.2.1.49.4.116.101.115.116 = INTEGER: 1
SNMPv2-SMI::mib-2.80.1.3.1.3.1.49.4.116.101.115.116 = STRING: "
ZZd"
SNMPv2-SMI::mib-2.80.1.3.1.4.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.5.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.6.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.7.1.49.4.116.101.115.116 = Gauge32: 0
SNMPv2-SMI::mib-2.80.1.3.1.8.1.49.4.116.101.115.116 = Gauge32: 0
```

(About the string displays "**ZZd**", please refer to **the notice** at **page 7** of this document.)

Step 4

⇨ **DGS-3620 starts to ping PC (10.90.90.100)**

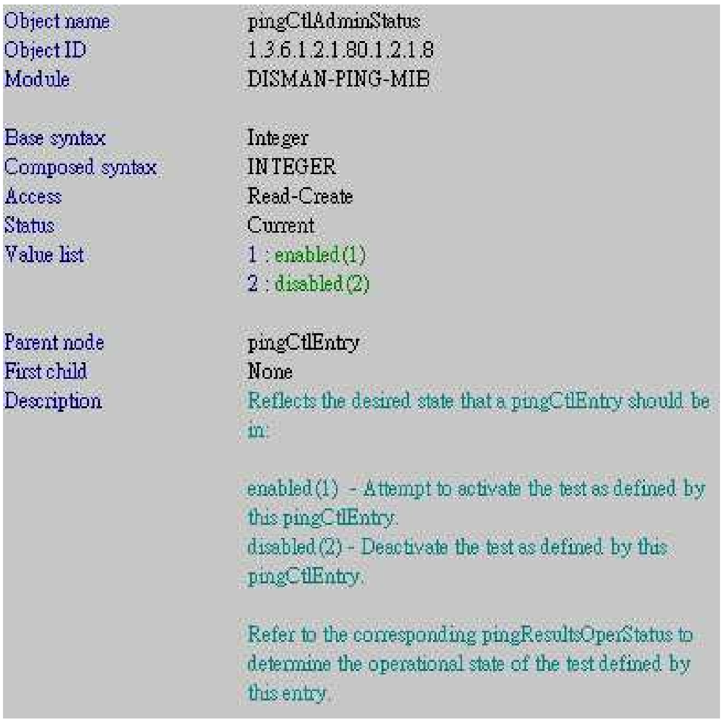


```
DGS-3620-28SC:admin#ping 10.90.90.100

Reply from 10.90.90.100, time<10ms
Reply from 10.90.90.100, time<10ms
Reply from 10.90.90.100, time<10ms

 Ping Statistics for 10.90.90.100
 Packets: Sent =3, Received =3, Lost =0
```

Step 5

⇨ **Enable Ping status before querying the result**

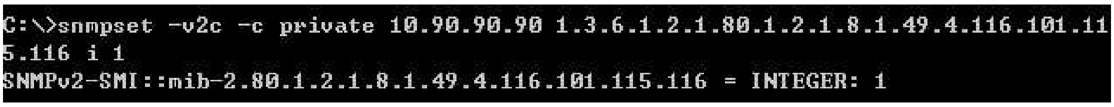
pingCtlAdminStatus (1.3.6.1.2.1.80.1.2.1.8)



Object name	pingCtlAdminStatus
Object ID	1.3.6.1.2.1.80.1.2.1.8
Module	DISMAN-PING-MIB
Base syntax	Integer
Composed syntax	INTEGER
Access	Read-Create
Status	Current
Value list	1 : enabled(1) 2 : disabled(2)
Parent node	pingCtlEntry
First child	None
Description	Reflects the desired state that a pingCtlEntry should be in: enabled(1) - Attempt to activate the test as defined by this pingCtlEntry. disabled(2) - Deactivate the test as defined by this pingCtlEntry. Refer to the corresponding pingResultsOperStatus to determine the operational state of the test defined by this entry.

To enable it by the command:

snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.8.1.49.4.116.101.115.116 ***i 1***



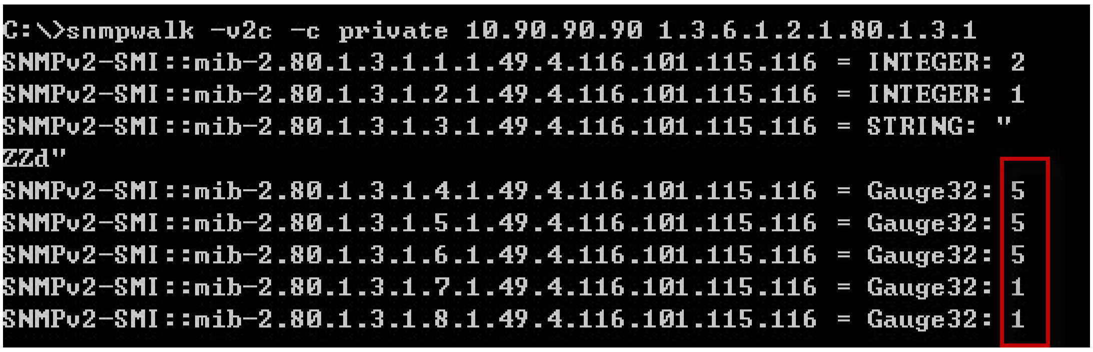
```
C:\>snmpset -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.2.1.8.1.49.4.116.101.11
5.116 i 1
SNMPv2-SMI::mib-2.80.1.2.1.8.1.49.4.116.101.115.116 = INTEGER: 1
```

Step 6

⇨ **Query the ping result**

To check the result by the command:

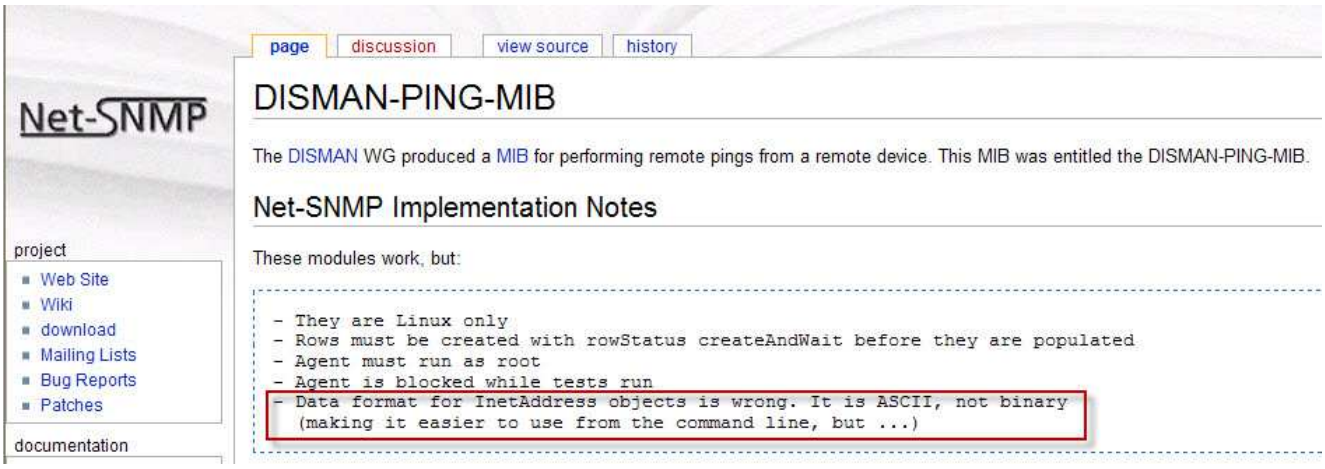
snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1



```
C:\>snmpwalk -v2c -c private 10.90.90.90 1.3.6.1.2.1.80.1.3.1
SNMPv2-SMI::mib-2.80.1.3.1.1.1.49.4.116.101.115.116 = INTEGER: 2
SNMPv2-SMI::mib-2.80.1.3.1.2.1.49.4.116.101.115.116 = INTEGER: 1
SNMPv2-SMI::mib-2.80.1.3.1.3.1.49.4.116.101.115.116 = STRING: "
ZZd"
SNMPv2-SMI::mib-2.80.1.3.1.4.1.49.4.116.101.115.116 = Gauge32: 5
SNMPv2-SMI::mib-2.80.1.3.1.5.1.49.4.116.101.115.116 = Gauge32: 5
SNMPv2-SMI::mib-2.80.1.3.1.6.1.49.4.116.101.115.116 = Gauge32: 5
SNMPv2-SMI::mib-2.80.1.3.1.7.1.49.4.116.101.115.116 = Gauge32: 1
SNMPv2-SMI::mib-2.80.1.3.1.8.1.49.4.116.101.115.116 = Gauge32: 1
```

Notice

We find there is a problem on Net-SNMP when it displays the Target IP address. Follow the standard (**RFC 4001**), we know to input the Target IP address by the hex value (0x0a5a5a64), but Net-SNMP displays this value by the ASCII code (0a=line feed, 5a=Z, 64=d). Hence we can see this string value is "ZZd". This issue has released on http://www.net-snmp.org/wiki/index.php/DISMAN-PING-MIB



DISMAN-PING-MIB

The DISMAN WG produced a MIB for performing remote pings from a remote device. This MIB was entitled the DISMAN-PING-MIB.

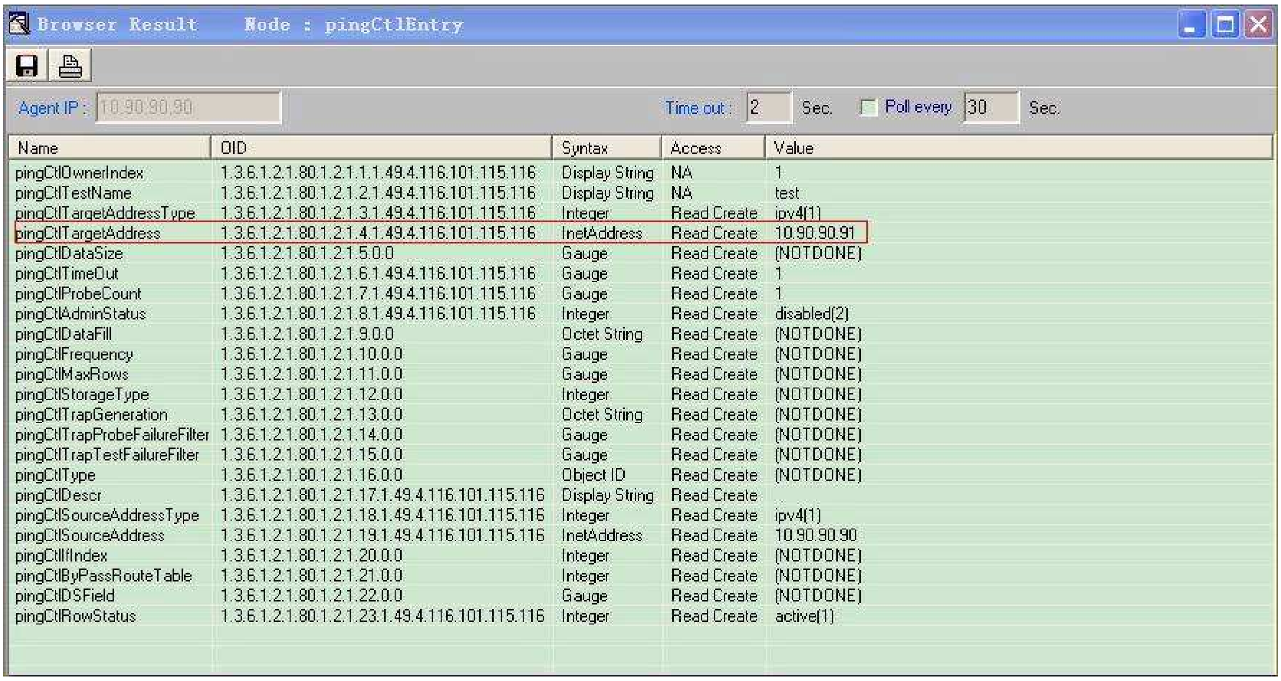
Net-SNMP Implementation Notes

These modules work, but:

```
- They are Linux only
- Rows must be created with rowStatus createAndWait before they are populated
- Agent must run as root
- Agent is blocked while tests run
- Data format for InetAddress objects is wrong. It is ASCII, not binary
  (making it easier to use from the command line, but ...)
```

We also query this value by some MIB Browsers, like **D-View Compiler** and **MG-SOFT**, and all works correctly,

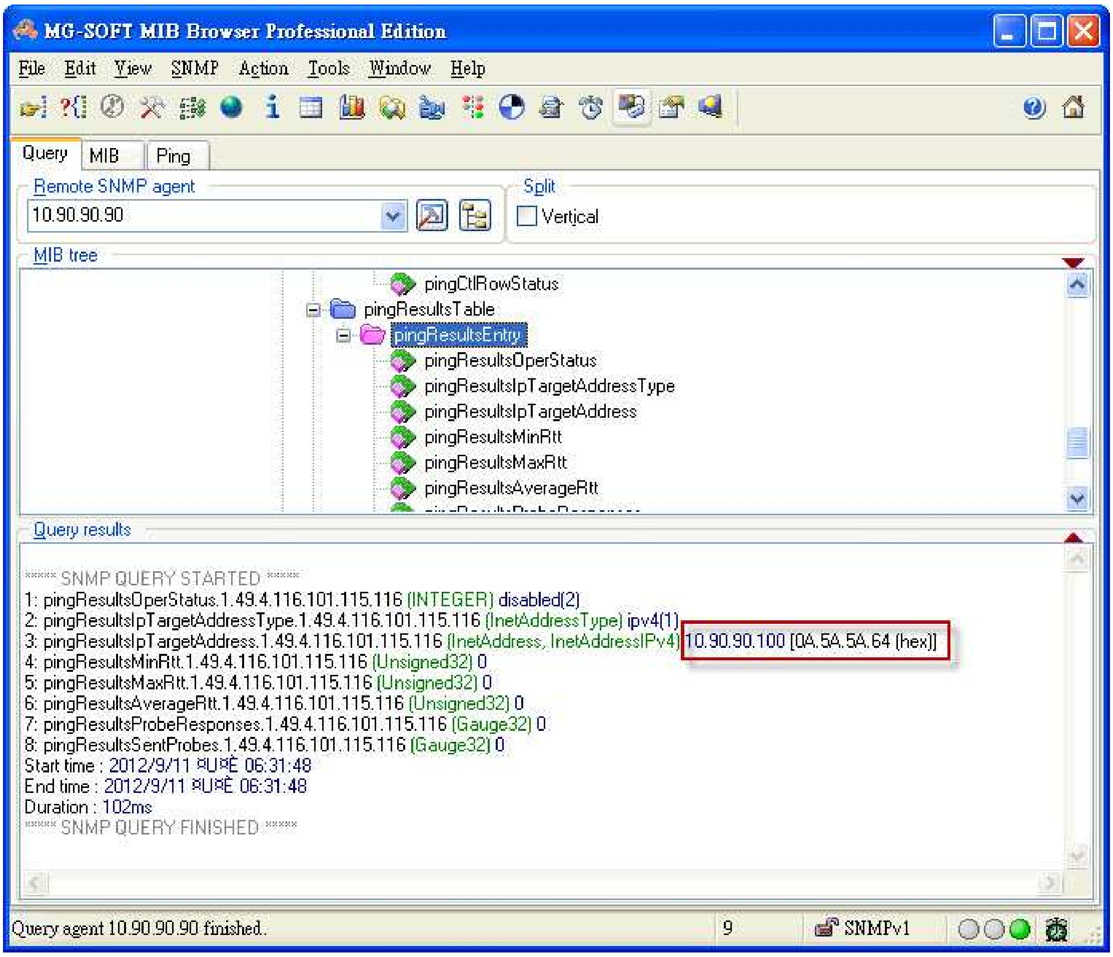
D-View Compiler (Target IP Address: 10.90.90.91)



Agent IP : 10.90.90.90   Time out : 2 Sec. □ Poll every 30 Sec.

Name	OID	Syntax	Access	Value
pingCtlOwnerIndex	1.3.6.1.2.1.80.1.2.1.1.1.49.4.116.101.115.116	Display String	NA	1
pingCtlTestName	1.3.6.1.2.1.80.1.2.1.2.1.49.4.116.101.115.116	Display String	NA	test
pingCtlTargetAddressType	1.3.6.1.2.1.80.1.2.1.3.1.49.4.116.101.115.116	Integer	Read Create	ipv4(1)
pingCtlTargetAddress	1.3.6.1.2.1.80.1.2.1.4.1.49.4.116.101.115.116	InetAddress	Read Create	10.90.90.91
pingCtlDataSize	1.3.6.1.2.1.80.1.2.1.5.0.0	Gauge	Read Create	(NOTDONE)
pingCtlTimeOut	1.3.6.1.2.1.80.1.2.1.6.1.49.4.116.101.115.116	Gauge	Read Create	1
pingCtlProbeCount	1.3.6.1.2.1.80.1.2.1.7.1.49.4.116.101.115.116	Gauge	Read Create	1
pingCtlAdminStatus	1.3.6.1.2.1.80.1.2.1.8.1.49.4.116.101.115.116	Integer	Read Create	disabled(2)
pingCtlDataFill	1.3.6.1.2.1.80.1.2.1.9.0.0	Octet String	Read Create	(NOTDONE)
pingCtlFrequency	1.3.6.1.2.1.80.1.2.1.10.0.0	Gauge	Read Create	(NOTDONE)
pingCtlMaxRows	1.3.6.1.2.1.80.1.2.1.11.0.0	Gauge	Read Create	(NOTDONE)
pingCtlStorageType	1.3.6.1.2.1.80.1.2.1.12.0.0	Integer	Read Create	(NOTDONE)
pingCtlTrapGeneration	1.3.6.1.2.1.80.1.2.1.13.0.0	Octet String	Read Create	(NOTDONE)
pingCtlTrapProbeFailureFilter	1.3.6.1.2.1.80.1.2.1.14.0.0	Gauge	Read Create	(NOTDONE)
pingCtlTrapTestFailureFilter	1.3.6.1.2.1.80.1.2.1.15.0.0	Gauge	Read Create	(NOTDONE)
pingCtlType	1.3.6.1.2.1.80.1.2.1.16.0.0	Object ID	Read Create	(NOTDONE)
pingCtlDescr	1.3.6.1.2.1.80.1.2.1.17.1.49.4.116.101.115.116	Display String	Read Create	
pingCtlSourceAddressType	1.3.6.1.2.1.80.1.2.1.18.1.49.4.116.101.115.116	Integer	Read Create	ipv4(1)
pingCtlSourceAddress	1.3.6.1.2.1.80.1.2.1.19.1.49.4.116.101.115.116	InetAddress	Read Create	10.90.90.90
pingCtlIfIndex	1.3.6.1.2.1.80.1.2.1.20.0.0	Integer	Read Create	(NOTDONE)
pingCtlByPassRouteTable	1.3.6.1.2.1.80.1.2.1.21.0.0	Integer	Read Create	(NOTDONE)
pingCtlDSField	1.3.6.1.2.1.80.1.2.1.22.0.0	Gauge	Read Create	(NOTDONE)
pingCtlRowStatus	1.3.6.1.2.1.80.1.2.1.23.1.49.4.116.101.115.116	Integer	Read Create	active(1)

MG-SOFT (Target IP Address: 10.90.90.100)



Remote SNMP agent: 10.90.90.90

Split: □ Vertical

```
***** SNMP QUERY STARTED *****
1: pingResultsOperStatus.1.49.4.116.101.115.116 (INTEGER) disabled(2)
2: pingResultsIpTargetAddressType.1.49.4.116.101.115.116 (InetAddressType) ipv4(1)
3: pingResultsIpTargetAddress.1.49.4.116.101.115.116 (InetAddress, InetAddressIPv4) 10.90.90.100 [0A.5A.5A.64 (hex)]
4: pingResultsMinRtt.1.49.4.116.101.115.116 (Unsigned32) 0
5: pingResultsMaxRtt.1.49.4.116.101.115.116 (Unsigned32) 0
6: pingResultsAverageRtt.1.49.4.116.101.115.116 (Unsigned32) 0
7: pingResultsProbeResponses.1.49.4.116.101.115.116 (Gauge32) 0
8: pingResultsSentProbes.1.49.4.116.101.115.116 (Gauge32) 0
Start time : 2012/9/11 ¤U¤È 06:31:48
End time : 2012/9/11 ¤U¤È 06:31:48
Duration : 102ms
***** SNMP QUERY FINISHED *****
```

Query agent 10.90.90.90 finished. 9 SNMPv1