

# **Event Monitor**

 this section is intended to be read in conjunction with the Introduction

# **Television Systems Limited.** Vanwall Road, Maidenhead, Berkshire, SL6 4UB Telephone +44 (0)1628 676200, FAX +44 (0)1628 676299

# **Event Monitor**

- 1.0 Introduction
- 2.0 Adding an Event
- 3.0 SNMP Trap only implemented on the product TMC-1
- 4.0 TSL UMD Input
- 5.0 Trigger Action

Appendix 1 – TSL MIB

These enhanced features are only available from TallyMan Version 1.51 (December 2006).

## 1.0 Introduction

This feature allows enhanced control of tallies and routers. In addition it is possible to upload a new TallyMan configuration file on receipt of an incoming trigger (tally/GPI/router change).

## 2.0 Adding an Event

Click on Add New Component on the main screen.

🔲 TallyMan - Untitled	
File Edit Tools View Comms	System Properties       Name:     New System       Platform:     Tallyman System Controller
	System Interfaces       Firmware         Add       Default System Interface       System:         Delete       Configure       Component Import/Export       V1.51         Add New Component       Name tally channels       Ø         Delete Component I       Configure       Configure
, Ready	

From the Add New Component list select Event Monitor.

Add New Sy Type:	Astem Component	Cance	When a new component is added to the system it must be given a Name before the OK button becomes active. This Name will be seen in the system tree on the left side of the screen.
Name:	Import	OK	

# Selecting the Event Monitor type

🔲 TallyMan - Untitled		
File Edit Tools View Comm	Event Monitor Properties	
	Name: Events Event type: SNMP v1 Trap ▼ SNMP v1 Trap TSL UMD input Trigger Action	]
	Edit <u>C</u> omms Parameters	
Ready		IUM //

Select the  $\ensuremath{\text{Event}}$  type from the drop down list.

# 3.0 SNMP Trap - only implemented on the product TMC-1

This description is intended as an overview for this facility which is currently (Nov 2007) only implemented on the TMC-1 product.

Simple Network Management Protocol

This is a communication protocol between management stations and managed objects, (such as routers and MDUs) and makes use of Management Information Bases (MIB) . SNMP uses a specified set of commands and queries. An MIB will contain information on these commands and on the target objects i.e. controllable entities or potential sources of status information.

A SNMP Event is shown as added to an existing system. Eight events have been added in this example.

TallyMan - Events.tms		)[	×
File Edit Tools View Comms	Help		~
New System     St System Tally     System Tally     System Tally     System C1     System C2     System C2	Event Monitor Properties		
Ready			1

Click on **Event** to open the list.

TallyMan - Events.tms						
<u>File E</u> dit <u>T</u> ools <u>V</u> iew <u>C</u> om	ms <u>H</u> elp					
🖃 🕎 New System	Index	Event	Mnemonic		Specific	
<mark>ST</mark> System Tally	1	Event 1	Over Temp			
🕂 🔣 Router C1	2	Event 2			(and a line of the	
🛨 🔣 Router C2	3	Event 3			8 <del></del> 8	
🛨 🔣 Router C5	4	Event 4			0 <del></del> 0	
🕀 📲 Router C6	5	Event 5				
🛨 🔝 Router C7	6	Event 6				
🛨 🎞 Parallel	7	Event 7				
🕂 📑 Display Ports	8	Event 8				
Events						
📈 Event						
	<				>	
Ready						

192 . 168 . 100 . 240
Enterprise OID:
rnet.private.enterprises.6853.2
Generic Trap type:
Enterprise Specific 📃
Specific Trap: 5

The Host IP address has been entered. This is the IP address of the unit that has to be monitored.

The Enterprise OID (Object IDentifiers) has to be entered; for TSL units this is 6853. The string: **.iso.org.dod.internet.private.enterprises**. (.1.3.6.1.4.1) is added automatically (presumed).

The string 6853 is unique to TSL.

If, say, a router is being monitored, the 6853.2 number shown above will be different, using the published numbers for that manufacturer's router. The 2 in this example would be the TSL MIB number.

The number currently published for TSL is actually only 6853

From the MIB: tslMIB OBJECT IDENTIFIER ::= { enterprises 6853 }

The generic trap type is shown as Enterprise Specific from the drop down list.

The **Specific Trap** will be the number that corresponds to the information required, as described in the MIB (Management Information Base). See Appendix 1 for the TSL MDU example.

Decide whether the Active Tally Channel should be Set or Clear for the tally/mnemonic.

#### Notes:

IANA, Internet Assigned Numbers Authority, assigns the IP and OID numbers.

http://www.iana.org/assignments/enterprise-numbers

and: http://www.alvestrand.no/objectid/1.3.6.1.4.html

The Firewall will need to be disabled (the factory default on a TMC-1) or the Ports 161/162 specifically opened.

When that Trap is sent, the mnemonic entered ( $\ensuremath{\text{Over Temp}}$  this example) will be shown on a UMD.

The UMD screen is shown below.

• Add the event to the UMD via a **System Tally**.

Allow user configuration		Name:		
Mapped Tallies In		Active Tally Channel Ma	sk	
Add Tally Delet	e Selection	Program	🗐 Iso 8	
	hannel	🔲 Iso 1	🗖 Iso 9	
	Program	🔲 Iso 2	🗖 Iso 10	
		🗐 Iso 3	☐ Iso 11	
		□ Iso 4	🗖 Iso 12	
		🔲 Iso 5	🖵 Iso 13	
		lso 6	🗖 Iso 14	
<b>K</b>	>	🗖 Iso 7	🗂 Iso 15	
Repeat Edit	Output Logic		anent On	Cancel

The use of a **System Tally** is essential for operation and also allow other **Events** to be added to the UMD.

Edit UMD 1 of umd			×
Name:	Brightness:	Normal	ОК
Display Text: Fixed: Fixed 1	– Display Tally: ––	Allow user configuration	Cancel
Display Assignment: Fixed Mnemonic			Restore Defaults
Matrix Recursion Depth: Maximum C Justify: Centre V Limited to: C 0			Configure Display
Matrix Assignment: Matrix: O (No Assignment)			Tally Bits
Direct Tally Assignment			
Type: ST System Tally  Parent: Parent: New System	Tally Channel Left Mask:		Repeat Edit
Tally: System Tally 1	Right Mask:		☐ Auto Inc ☐ Auto Copy

This is a very specific use of the UMD. The text and Tally Lights will change in the event of a trap being received.

• Write the file to the TMC-1 and go on-line.

Correct communications will show by a green dot against the **Events** entry when on-line, as is normal.

## 4.0 TSL UMD Input

🔲 TallyMan - Untitled		
<u>File E</u> dit <u>T</u> ools ⊻iew <u>C</u> omms	Help	
Performance in the system	Event Monitor Properties          Name:       Events         Event type:       TSL UMD input         Number of Events:       2         Edit Comms Parameters	2
Ready		NUM /

**Edit Comms Parameters** for the data incoming port into the TallyMan unit. The TallyMan controller will expect standard TSL UMD Protocol.

Incoming tallies start at Address 000. Two are shown above so they will be at address 000 and 001. These may then be mapped to control a tally lamp on any UMD or may be output to the Tally Output pins etc.

Events: Setup Communication			? 🛛
Type: Serial RS422			OK Cancel
General Parameters Port Number: 1	Serial Parameters Baud Rate:	38400	<b>_</b>
Description: Event Input	Parity:	Even	•
Network Parameters	Data Bits:	8	•
IP Address:	Stop Bits:	1	•

#### Select the Event.

🔲 TallyMan - Untitled					
<u>File E</u> dit <u>T</u> ools <u>V</u> iew <u>C</u> on	nms <u>H</u> elp				
🖃 🕎 New System	Index	Event	Mnemonic	Specific	
Events	1	Event 1	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	No Tally	
No Event	2	Event 2		No Tally	
Ready	1			-	

An incoming tally will show its status in the Specific column.

The incoming tally signal may be set to any Channel. Please see the Tally Section for more information.

Edit Event 1 of Events				X
Name:				OK Cancel
Tally Channel (Left)		Tally Channel (Right)		
✓ Program	☐ Iso 8	Program	🗔 Iso 8	
🔽 Iso 1	🔲 Iso 9	🗖 Iso 1	🗐 Iso 9	
🔽 Iso 2	🔲 Iso 10	🗖 Iso 2	🔲 Iso 10	
🖵 Iso 3	🔲 Iso 11	🗖 Iso 3	🗔 Iso 11	
🔽 Iso 4	🔲 Iso 12	🔲 Iso 4	🗔 Iso 12	
🔲 Iso 5	🔲 Iso 13	🔽 Iso 5	🔲 Iso 13	
🖵 Iso 6	🔲 Iso 14	🖵 Iso 6	🗔 Iso 14	
🔲 Iso 7	🔲 Iso 15	🗖 Iso 7	🔲 Iso 15	
Repeat Edit Auto Inc Auto Copy	Allow user configuration			

This may then be mapped to a tally function such as activating a tally LED on a UMD.

Edit UMD 1 of UMD		X
Name: Display Text:	Brightness: Normal 💽 Allow user configuration 🦵	OK Cancel
Fixed: Fixed 1 Display Assignment: Fixed Mnemonic	- Display Tally:	Restore Defaults
Matrix Recursion Depth: Maximum C Justify: Centre  Limited to: C		Configure Display
Matrix: Contraction Assignment Contraction		
Direct Tally Assignment		
Type: Event  Parent: ?! Events Talks	Tally Channel: Left Mask: Right Mask:	Repeat Edit
Tally: Event 1		Auto Copy

Note: The correct Tally Channel mask must be set in the usual way.

# 5.0 Trigger Action

TallyMan - TM2 test.tms	
<u>File E</u> dit <u>T</u> ools <u>V</u> iew <u>C</u> omms	Help
New System	Event Monitor Properties          Name:       Events         Event type:       Trigger Action         Number of Events:       2         Edit Comms Parameters
Ready	- OFFLINE NUM

There is no interface or port to be set with this event.

Events: Setup Communication		? 🛛
Type: No Interface		OK Cancel
General Parameters	Serial Parameters	
Port Number: 0	Baud Rate:	<u>~</u>
Description:	Parity:	<u></u>
Network Parameters	Data Bits:	~
IP Address: An An An	Stop Bits:	<b>_</b>

## Select the Event.

Name:	Allow user configuration	Cano
igger Type:	Action Type:	OK
No Trigger 🗨	No Action	
No Trigger Tally On Tally Off TallyChange Single Route	Edit Output Tally	
Router Source:	Source:	
Routed to Dest:	Routed to Dest:	
	No File	_ Repeat Edit

Say we are going to use a Tally (or incoming GPI) to trigger an event.

ction Type:	OK
	UK
No Action	
Edit Output Tally	
Source:	
Routed to Dest:	
No File	
	Edit Output Tally

Select the required input tally from the available list.

Type: Parent: Tally:		· 	Tally 노토 Tally In 1	Parent Parallel	Logic
-Logic- Opera ∏ In		> Add >			
Mode	<ul> <li>Bitwise (per channel)</li> </ul>	]			

		Tally	Parent	Logic
Type: 🔤 Tally In		노 <mark>I</mark> Tally In 1	Parallel	
Parent: TO Parallel				
Tally: Tally In 1				
Logic	> Add >			
Operator: Or				
Invert				
Mode				
C Logical (any channel)	Finished		Delete Selection	1

Now select the required Action.

igger	Allow user configuration	Cano
Туре:	Туре:	OK
Tally On 💌	No Action	
Edit Input Tally	T ally Active Set Tally Clear Tally Route Load File Reverse route Copy Route T able Route T opgle Tally Routed to Dest: 3€	
	No File	at Edit

No Action:	Nothing will happen.
Tally Active:	The tally will follow the Trigger status, i.e. ON/OFF.
Set Tally:	Tally is turned ON (remains ON even when the trigger has gone).
Clear Tally:	Tally is turned OFF (remains OFF even when the trigger has gone).
Route:	A specific router crosspoint will be made.
Load File:	A Router file (.rtr) or a TallyMan File (.tms) will be loaded (Care!).
Reverse route:	Inputs/Outputs are reversed on the controlled router.
Copy Route:	Routers are effectively paralleled for crosspoint control.
Table Route:	This allow control whereby when one destination is switched, another
	destination is switched as a slave. This may be on the same or on a
	different router.
Toggle Tally:	With an incoming tally, say, the tally output will toggle on and then off
	with the next action.

# Tally Active/Set Tally/Clear Tally

Edit Tally Mask		X
Select active tally channel	for event:	OK Cancel
Tally Channel		
Program	🖵 Iso 8	
🗖 Iso 1	🔲 Iso 9	
🗖 Iso 2	🔲 Iso 10	
🔲 Iso 3	🔲 Iso 11	
🗖 Iso 4	🔲 Iso 12	
🗖 Iso 5	🗖 Iso 13	
🗖 Iso 6	🔲 Iso 14	
🔲 Iso 7	🔲 Iso 15	
Assigned to output tally o	of Event: This event	•
	This event Event 1	

The **Assigned to output tally of Event** is an option for the Events configured.

You may, for example, set a tally with Event 1 and turn this tally off in Event 1 using action from Event 2.

🔲 TallyMan - Event TM2	. tms					
<u>Eile E</u> dit <u>T</u> ools <u>V</u> iew <u>C</u> om	ims <u>H</u> elp					
🖃 🕎 New System	Index	Event	Trigger Type	Action Type	Action Data	Output Assignment
🕂 📲 Router C1	1	Event 1	Tally On	Set Tally	1: Program	This Event
🕂 📲 Router C2	2	Event 2	Tally On	Clear Tally	1: Program	Event 1
🕂 🗐 Router C5						
🕂 📲 Router C6						
🕂 📲 Router C7						
⊕ <mark>표위</mark> Parallel						
庄 🚟 Display Ports						
Events						
📈 📈 Event						
Des de	1				- the second sec	
Ready					📲 🗐 OF	FLINE NUM

Name:	Allow user configuration	Cano
gger	Action	04
Type:	Type:	
Tally On 💌	Route	
Edit Input Tally	Edit Output Tally	
<b></b>		
Source:	Src 1	
Routed to Dest:	Dst 1	
	No File	Repeat Edit
	Dst 1	

You may use a Tally ON for example to make a router crosspoint.

Clicking the buttons will call up the router lists.

Source Man	ager				×
Matrix: Category:	Router C1			• <u>N</u> ame • <u>M</u> nemo	1000
Source			emonic	View D	etails Mix 🔨
Source 1	1	Src			
Source 2	2		2		
Source 3	3	Src	3		
Source 4	4	Src	4		~
<	1111				<b>&gt;</b> .::

Double click on the required Source or Destination.

#### Load File.

This will load the router  ${\boldsymbol r} {\boldsymbol t} {\boldsymbol r}$  status file . or a new .  ${\boldsymbol t} {\boldsymbol m} {\boldsymbol s}$  Config file.

Note: With the .**tms** file you need to be sure that you have programmed the new file with the appropriate module in order to return to the original file if you wish to retain this automation.

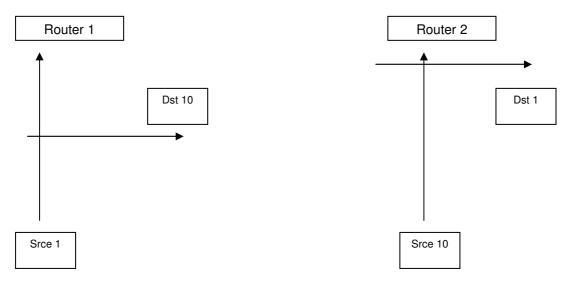
Edit Event Action 1 of Events		
Name: Trigger Type: Tally On	Allow user configuration Action Type: Load File	Cancel OK
Edit Input Tally	Edit Output Tally	
Routed to Dest:	Routed to Dest:	Repeat Edit

: Load Setup file:	: Load Setup file:	? 🛛
Look in: 🔁 TallyMan 💽 🔶 🖻 👘 🏢	Look in: 🧰 TallyMan 💌 🔶 🖻	• 📰 🎽
© one.rtr	Image: 24_7v1.tms     Image: 25P-1TM1setup.tms       Image: 24seven v2.tms     Image: 24seven v2.tms       Image: 24seven v2.tms     Image: 24seven v2.tms <tr< th=""><th>W 58C test.tms Simple imd.tms TM1 final file.tm TM1 IMD Test. TM1 SC8 Test. TM1 test.tms</th></tr<>	W 58C test.tms Simple imd.tms TM1 final file.tm TM1 IMD Test. TM1 SC8 Test. TM1 test.tms
File name:     One.ttr       Files of type:     Routing files (".ttr)	File name:     one.rtr       Files of type:     System files (".tms)	<u>O</u> pen Cancel

# Reverse Route

Name:	Allow user configuration	Cancel
Frigger Type:	Action Type:	(OK
Single Route	Reverse route	
Edit Input Tally	Router C2	
<b></b>	<b>_</b>	
Src 1	Source:	
Dst 10	Routed to Dest:	
	No File	Repeat Edit
		T Auto Inc

In the case shown, when Router 1 has Source 1 routed to Destination 10, Router 2 will have Source 10 routed to Destination 1.



Name:	Allow user configuration	Canc
gger Type:	Action Type:	OK
Router	Reverse route	
Router C1	Router C2	
	<u></u>	
Source:	Source:	
Routed to Dest:	Routed to Dest:	
	No File	– Repeat Edit

This will reverse route the second Router C2 completely.

#### Copy Route

This effectively parallels router control.

Router 1: Source 1 is routed to Dest 10 Router 2: Source 1 is routed to Dest 10.

#### Table Route

This allow control whereby when one destination is switched, another destination is switched as a slave. This may be on the same or on a different router.

Name: Event 1		Allow user config	guration Can
rigger Type:		Action Type:	
Router	•	Table Route	- -
probel 1		Set Router	
	<b>*</b>	J	<u>×</u>
Source:	-98	Source:	<u>88</u>
Routed to Dest:	<u>18</u>	Routed to Dest:	<del>18</del>
		Edit Table	Repeat Edit

• Select Edit Table

et Router Destination Link			Set Router Source Link			Cance
rigger Dest	Action Dest		Trigger Srce	Action Srce		
1: Destination 1 2: Destination 2 3: Destination 3			1: Source 1 2: Source 2 3: Source 3		^	
4: Destination 4 5: Destination 5			4: Source 4 5: Source 5			Action router active leve
6: Destination 6 7: Destination 7 8: Destination 8			6: Source 6 7: Source 7 8: Source 8			Single Level Router
9: Destination 9 10: Destination 10 11: Destination 11			9: Source 9 10: Source 10 11: Source 11			
12: Destination 12 13: Destination 13			12: Source 12 13: Source 13			
14: Destination 14 15: Destination 15 16: Destination 16		~	14: Source 14 15: Source 15 16: Source 16		~	Preset 1:1
Edit 1: Destination 1			⊢ Edit Source Assignment			Clear All
Clear		•			<b>T</b>	
 	OK		T AutoInc	OK		

Preset 1: 1 will match each destination and sources with their counterpart or individual destinations to other destinations etc. may be mapped.

1:1 setting selected.

Edit Table Routing A	ssignment				
Set Router Destination			Set Router Source Link		Cancel
Trigger Dest	Action Dest		Trigger Srce	Action Srce	
1: Destination 1 2: Destination 2 3: Destination 3 4: Destination 4 5: Destination 6 7: Destination 6 7: Destination 7 8: Destination 7 9: Destination 9 10: Destination 10 11: Destination 11 12: Destination 12 13: Destination 14 15: Destination 15 16: Destination 16	Destination 33 Destination 34 Destination 35 Destination 36 Destination 37 Destination 38 Destination 40 Destination 40 Destination 41 Destination 42 Destination 43 Destination 45 Destination 46 Destination 47 Destination 48		1:Source 1 2:Source 2 3:Source 2 3:Source 4 5:Source 5 6:Source 5 7:Source 7 8:Source 9 10:Source 9 10:Source 9 10:Source 10 11:Source 11 12:Source 11 12:Source 13 14:Source 15 16:Source 16	Source 33 Source 34 Source 35 Source 36 Source 37 Source 38 Source 39 Source 40 Source 41 Source 41 Source 41 Source 44 Source 44 Source 45 Source 45 Source 47 Source 48	OK Action router active levels: Level: Single Level Router
Edit 1: Destination 1	Destination 33	•	Edit Source Assignme	ntOK	Clear All

The two routers are shown. One starts with the system destination number as Destination 1 and the second router is shown with the router destination set as Destination 33.

Individual Destinations to destinations selected.

et Router Destination I			Set Router Source Link	200 generation 200 million		Cancel
Frigger Dest	Action Dest		Trigger Srce	Action Srce		
1: Destination 1	Destination 34	~	1:Source 1	Source 38	~	OK
2: Destination 2			2: Source 2			
3: Destination 3			3: Source 3			Action router active level
4: Destination 4			4: Source 4			Action fouter active level
5: Destination 5			5: Source 5	1.000		Level:
6: Destination 6			6: Source 6			Single Level Router
7: Destination 7			7: Source 7			Single Level Houler
8: Destination 8			8: Source 8			
9: Destination 9			9: Source 9			
10: Destination 10			10: Source 10			
11: Destination 11			11: Source 11			
12: Destination 12			12: Source 12			
13: Destination 13			13: Source 13			
14: Destination 14			14: Source 14			
15: Destination 15			15: Source 15			
16: Destination 16	6770	~	16: Source 16		~	Preset 1:1
Edit 1: Destination 1-			Edit 1: Source 1			Clear All
<u>C</u> lear	Destination 34	•	Clear	Source 38	-	
EAL	OK		E ALL	OK		
AutoInc	OK		Auto <u>I</u> nc	L	i	

#### Appendix 1

TSL-MIB DEFINITIONS ::= BEGIN IMPORTS enterprises, Opaque FROM RFC1155-SMI OBJECT-TYPE FROM RFC-1212 TRAP-TYPE FROM RFC-1215; -- MODULE-IDENTITY -- FROM SNMPv2-SMI; -- TSL\_MIB; SNMP v1 agent definitions. -- the following only allowed in SMIv2 (also 0 enumeration of integers) -- As of 08/08/03, includes enterprise specific trap definitions (RFC1215) -- tslMIB MODULE-IDENTITY LAST-UPDATED "0308080000Z" \_\_\_ ORGANIZATION "Television Systems Ltd" \_\_\_ \_\_\_ CONTACT-INFO " \_\_\_ Tim Whittaker \_\_\_ Television Systems Ltd \_\_\_ Unit 4, King's Grove \_\_\_ Maidenhead \_\_\_ Berkshire \_\_\_ SL6 4DP \_\_\_ \_\_\_ Tel + 44 1628 687200 \_\_\_ Email: timw@televisionsystems.ltd.uk" DESCRIPTION "MIB module for all TSL products" \_\_\_ \_\_\_ **::=** { enterprises 6853 } DisplayString ::= OCTET STRING -- SMIv1 definition of module tslMIB OBJECT IDENTIFIER ::= { enterprises 6853 } ----- Winsoft specific MIB -- DELETED for mdu12 hardware \_\_\_\_\_ ----- generic alarm MIB (all TSL equipment capable of SNMP alarms) alarm OBJECT IDENTIFIER ::= { tslMIB 2 } alarmIdent OBJECT-TYPE SYNTAX DisplayString ACCESS read-only

```
STATUS mandatory
    DESCRIPTION
    "Equipment alarms description and version"
    ::= { alarm 1 }
----- alarm table
alarmTable OBJECT-TYPE
    SYNTAX SEQUENCE OF AlarmEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
    "The table of alarm entries"
    ::= { alarm 2 }
alarmEntry OBJECT-TYPE
    SYNTAX AlarmEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
    " An alarm entry in the table"
     INDEX { alarmTableIndex }
    ::= { alarmTable 1}
AlarmEntry ::= SEQUENCE
      {
            alarmTableIndex INTEGER,
            alarmType INTEGER,
            alarmIndex INTEGER,
            alarmText DisplayString,
            alarmState INTEGER,
            alarmPolarity INTEGER,
            alarmData Opaque
      }
alarmTableIndex OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "The index into the table"
    ::= { alarmEntry 1 }
alarmType OBJECT-TYPE
    SYNTAX INTEGER
    {
            internal(1), -- general internal to equipment alarm
                                    -- from external GPI,
            gpi(2),
alarmPolarity determines alarmState
            outputFail(3), -- eg MDU12 output fuse, etc
                              -- alarmData is text describing failure
            psuFail(4)
(eg rail values etc)
   }
    ACCESS read-only
   STATUS mandatory
    DESCRIPTION
    "Alarm type"
    ::= { alarmEntry 2 }
```

```
alarmIndex OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "Alarm type number"
    ::= { alarmEntry 3 }
alarmText OBJECT-TYPE
    SYNTAX DisplayString
   ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "Alarm description"
    ::= { alarmEntry 4 }
alarmState OBJECT-TYPE
   SYNTAX INTEGER
    {
           inactive(1),
        active(2)
    }
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "Alarm state"
    ::= { alarmEntry 5 }
alarmPolarity OBJECT-TYPE
   SYNTAX INTEGER
    {
           notApplicable(1),
        normallyOpen(2),
        normallyClosed(3)
    }
   ACCESS read-write
   STATUS mandatory
   DESCRIPTION
    "Alarm active polarity (notApplicable for non-gpi alarms)"
    ::= { alarmEntry 6 }
alarmData OBJECT-TYPE
   SYNTAX Opaque
   ACCESS read-only
   STATUS optional
   DESCRIPTION
   "Additional alarm data of variable length, according to alarm
type."
   ::= { alarmEntry 7 }
----- end of table
alarmTotal OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-only
   STATUS mandatory
   DESCRIPTION
    "The total number of alarms in the table"
    ::= { alarm 3 }
```

```
alarmLocation OBJECT-TYPE
   SYNTAX DisplayString
   ACCESS read-write
   STATUS mandatory
   DESCRIPTION
    "The physical location of the equipment generating the alarm"
    ::= { alarm 4 }
alarmEqptTemp OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS optional
   DESCRIPTION
    "Equipment temperature (in degrees Centigrade)"
    ::= { alarm 5 }
alarmEqptTempHi OBJECT-TYPE
   SYNTAX INTEGER
   ACCESS read-write
   STATUS optional
   DESCRIPTION
    "Equipment temperature alarm point (degrees Centigrade)"
    ::= { alarm 6 }
alarmTrap TRAP-TYPE
     ENTERPRISE tslMIB
     VARIABLES
      {
           alarmTableIndex,
           alarmType,
           alarmIndex,
           alarmText,
           alarmState,
           alarmPolarity,
           alarmData
      }
      DESCRIPTION
      "An entry in the alarm table has changed state"
      ::= 4
alarmEqptTempHiTrap TRAP-TYPE
      ENTERPRISE tslMIB
      VARIABLES
      {
           alarmEqptTemp
      }
      DESCRIPTION
      "The equipment temperature has exceeded the maximum allowed"
      ::= 5
alarmEqptTempOkTrap TRAP-TYPE
     ENTERPRISE tslMIB
      VARIABLES
      {
           alarmEqptTemp
      }
      DESCRIPTION
      "The equipment temperature is now within limits"
      ::= 6
                  _____
```

```
----- MDU12 specific MIB
mdu12 OBJECT IDENTIFIER ::= { tslMIB 3 }
mdu12Ident OBJECT-TYPE
   SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Equipment description and version"
    ::= { mdu12 1 }
mduPowerOn OBJECT-TYPE
    SYNTAX INTEGER
      {
            simultaneous(1),
            sequential(2),
            delayed(3)
      }
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "MDU power-on output sequence"
    ::= { mdu12 2 }
mduSeqDelay OBJECT-TYPE
   SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "Sequential mode delay between outputs"
    ::= { mdu12 3 }
mduOutputTable OBJECT-TYPE
    SYNTAX SEQUENCE OF MduOutputEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
    "Table of output controls"
    ::= { mdu12 4 }
mduOutputEntry OBJECT-TYPE
    SYNTAX MduOutputEntry
   ACCESS not-accessible
   STATUS mandatory
    DESCRIPTION
    " An entry in the output table"
    INDEX {mduOutputIndex}
    ::= { mduOutputTable 1}
MduOutputEntry ::= SEQUENCE
      {
       mduOutputIndex INTEGER,
        mduOutputState INTEGER,
        mduOutputDelay INTEGER
      }
mduOutputIndex OBJECT-TYPE
```

```
SYNTAX INTEGER(1..12)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
    "Output number"
    ::= { mduOutputEntry 1 }
mduOutputState OBJECT-TYPE
   SYNTAX INTEGER
     {
            off(1),
            on(2),
           locked-Off(3), -- locked by admin web page, cannot
change via SNMP
           locked-On(4) -- locked by admin web page, cannot
change via SNMP
     }
   ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "MDU Output status"
    ::= { mduOutputEntry 2 }
mduOutputDelay OBJECT-TYPE
   SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
    "Output on delay from power on (Delay mode only)"
    ::= { mduOutputEntry 3 }
-- End of table
mduPowerStatus OBJECT-TYPE
   SYNTAX INTEGER
    {
           totalLoss(1),
           input10K(2),
           input20K(3),
           allOk(4)
                          -- note: a single input MDU would
report allOk if power is present
     }
    ACCESS read-only
    STATUS mandatory
   DESCRIPTION
    "Status of MDU power inlets"
    ::= { mdu12 5 }
mduPowerStatusTrap TRAP-TYPE
      ENTERPRISE tslMIB
      VARIABLES
      {
           mduPowerStatus
      }
      DESCRIPTION
      "The power input to the MDU has changed state"
      ::= 7
  _____
```

```
END
```