



**UTMC**

Urban Traffic Management and Control

**TS004.001:2005**  
**UTMC Objects Registry**

May 2005

Cover + 99 pages

© Crown Copyright 2005

## Foreword

This document, UTMC Technical Specification 004 version 001 (TS004.001:2005), was prepared by UTMC Programme Management and is published under the authority of the Department for Transport (DfT). It forms part of the range of UTMC specifications and supersedes previous versions of the Technical Specification, TS001:1997 and TS002:2001.

TS004.001:2005 takes into account feedback from the consultation process and ongoing international initiatives in this field. It presents the data standards recommended for use by UK traffic managers in their systems. Details are provided in normative annexes.

This document should be used in conjunction with the other main repository of UTMC technical recommendations, namely the Framework Technical Specification, TS003. TS004 is under continuous review and update, while TS003 is not likely to be updated for 2-3 years.

Copies of all UTMC documentation, together with background material and other information, can be found on the UTMC website at: <http://www.utmc.gov.uk>

**Please note:** (1) Compliance with this specification does not of itself confer immunity from or compliance with any legal obligations. (2) Whilst DfT strongly supports the adoption of UTMC standards, such standards are not mandatory.

The Department for Transport and the UTMC Development Group gratefully acknowledges the considerable amount of work contributed to the development of this specification, and of products that conform to it, by the members of the UTMC Suppliers Forum.

## List of contents

<b>Foreword</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 General	3
1.2 Document approach and structure	3
1.3 Relationship to prior documents	4
<b>2 Object registration</b>	<b>5</b>
2.1 Registered Object approval status	5
2.2 Key process: registration of an Object	5
<b>A References (Normative)</b>	<b>7</b>
<b>B Management authority for this document (Informative)</b>	<b>8</b>
B.1 Formal authority	8
B.2 Day to day management	8
<b>C Schedule of registered objects (Informative)</b>	<b>9</b>
C.1 Registered Objects	9
<b>D Definitions of registered Data Objects (Normative)</b>	<b>11</b>
<b>E Definitions of registered MIBs (Normative)</b>	<b>12</b>
E.1 Introduction	12
E.2 UM/003, UTMC header MIB	12
E.3 UM/004, Air quality monitor MIB	15
E.4 UM/006, VMS MIB	30
E.5 UM/015, Simple UTC MIB	42
E.6 UM/016, UTC MIB	48
E.7 UM/029, Car Park Monitor MIB	68
E.8 UM/030, Traffic Counter MIB	83
<b>F Definitions of registered IDL scripts (Normative)</b>	<b>98</b>
<b>G Definitions of registered Other Objects (Normative)</b>	<b>99</b>

# 1 Introduction

## 1.1 General

- 1.1.1 TS004 provides standards for UTMC “common data” (ie data communicated between applications of a UTMC system, or between a UTMC system and an external system) through holding definitions of current UTMC Data Objects and session management protocols, and making them available to users.
- 1.1.2 A new UTMC system should normally use the currently registered UTMC Data Objects and session management protocols as far as they are applicable. Where local needs require the development of new Data Objects and session management protocols, these should normally be submitted for approval to the Registry. This will ensure that any potential overlaps (for example with Data Objects currently under review for registration) are identified.
- 1.1.3 Anyone can make free use of the Registered Objects, whether system/software developer or user. Users may freely specify the use of particular Registered Objects in procuring systems. There is no obligation arising upon use.
- 1.1.4 This document is designed for use alongside TS003:2005, the UTMC Technical Specification, which provides a framework of general standards for UTMC systems.

## 1.2 Document approach and structure

- 1.2.1 TS004 is composed of numbered clauses and subclauses, which form the normative elements of the specification. The titles of each clause are listed in the contents list. This document incorporates, by reference, provisions from specific editions of other publications (Normative references) and other publications that provide information or guidance (Informative references). These references are cited at the appropriate points in the text.
- 1.2.2 The following annexes are included. Annexes A, D, E, F and G are normative and form an integral part of this specification. Other annexes are informative and merely provide guidance on how the normative aspects of this document could be used.

Annex A: Normative Reference Documents: a list of all normative documents referenced in this document.

Annex B: Management authority for this document: a statement of how this document is maintained and where questions relating to its provisions or updates should be addressed.

Annex C: Schedule of registered objects and contact points: a list of the objects defined in this specification together with information on the sponsors and contexts of the object sponsorship.

Annex D: Definitions of registered Data Objects: the specification of objects to be used in UTMC common databases and for inter-application interfaces in a UTMC system.

Annex E: Definitions of registered MIBs: the specification of MIB objects to be used between UTMC components where SNMP is the chosen data exchange protocol.

Annex F: Definitions of registered IDL scripts: the specification of IDL modules to be used in the context of a CORBA interface.

Annex G: Definitions of registered Other Objects: the specification of objects to be used under circumstances other than those in Annexes D, E and F. There are currently no entries in this Annex.

1.2.3 Annexes D and F are presented in the form of spreadsheet workbooks separate from this document.

### 1.3 Relationship to prior documents

1.3.1 Through the course of the UTMC programme the Objects Registry has been developed as an evolving library of specification components. A number of these have been documented in project reports, were used in products, and are now operational in live systems. TS004 builds on all of these.

1.3.2 However the primary function of TS004.001 is to set a clear, coherent starting point for UTMC as it enters the mainstream. As a result, **there is no guarantee of consistency between TS004.001:2005 and prior documents.** This position has been the subject of widespread consultation and has been accepted – although inevitably it does mean that users of pre-existing documents and systems will need to consider whether they need to review their UTMC compliance against this document.

## 2 Object registration

### 2.1 Registered Object approval status

2.1.1 A UTMC Registered Data Object may pass through a number of stages:

- a) *Notification.* A Proposer has notified the Data Registry of intention to develop and register an Object.
- b) *Submission.* The Object has been developed and its definition submitted to the Data Registry.
- c) *Validation.* The Object has been validated in a tested implementation.
- d) *Establishment.* The Object has been approved following public consultation.
- e) *Retirement.* The Object is formally withdrawn from active promotion.

2.1.2 An Object will not necessarily pass through all these stages.

2.1.3 Objects are listed as UX/nnn, where X is C for a Data Object, M for a MIB Object, I for an IDL Object and O for any other registered Object (specifically including XML Internet Objects) with nnn denoting the reference number allocated to the Object by the Registry.

### 2.2 Key process: registration of an Object

2.2.1 Objects are registered by a Proposer, which might be a local authority, a systems developer, or a consultancy. Intellectual Property Rights in the published aspects of Registered Objects are held by the Crown.

2.2.2 In some cases suppliers may elect to register an Object themselves. In other cases a local authority may choose, during the course of sponsoring a system development, to require suppliers to register relevant Objects.

2.2.3 To take an Object through Registration requires the submission of documentation to the Data Registry. The documentation required will not normally be anything additional to what good design practice would ordinarily require, so should not be expensive. Documentation should be submitted in suitable electronic form, ie normally in a text file for a MIB and a spreadsheet, using the template provided, for a database object.

2.2.4 The proposed process of registration is as follows:

- a) *Notification.* The Proposer proposes an Object with a name, brief description of purpose and content, applicable technology and any other key initial information. He also sets a date by which he will have it developed; this will not normally be further than one year from the date of Notification. The Object is given a reference number by the Registry, and short details published. Included is contact information for the Data Object developer. This is included to facilitate communication between interested parties, as other UTMC implementers may wish to ensure that aspects relevant to their work are included in the developing Object. Also, where two similar Data Objects are registered, developers might wish (and will be encouraged) to pool resources and submit a joint Data Object. As UTMC is an open specification, this is a key aspect of the notification stage.

- b) *Submission.* The Object is submitted to the Data Registry in a suitable formal language. The submission should include proposed test criteria for validation. It is given a version number and made available publicly for testing. The Data Object definition is placed on the UTMC website, typically as a downloadable excel file.
- c) *Validation.* The Object must have been implemented and tested. Evidence should be provided to the Data Registry that the Submitted Object was tested against a relevant set of functional and performance criteria. It should also be tested for compatibility with other relevant Validated or Established Objects, though this might be done as a paper exercise rather than in a system implementation.
- d) *Establishment.* Once validated, the Proposer may request that the Data Registry seek Establishment. This should be undertaken by consultation (formal responses requested from suitable referees).
- e) *Retirement.* The Data Registry may withdraw an Established Object, normally with the Proposer's consent, in the event that it is superseded or is no longer applicable for other reasons. The Retired Object definition will remain publicly accessible.

2.2.5 It is possible for multiple overlapping Objects to be Notified or Submitted. The managers of the Registry may discuss whether it is appropriate to remove overlaps at Validation, and this will be a specific query in consultation for Establishment.

## A References (Normative)

A.1 The following is a list of documents to which normative reference is made in the main text of this document in such a way as to make them indispensable for the application of the standard.

- a) Road Traffic Regulation Act 1988
- b) Traffic Signs Regulations and General Directions : 1994
- c) Pelican Pedestrian Crossings Regulations and General Directions : 1987
- d) MCH 1813 : Highways Agency
- e) MCE 360C : Highways Agency
- f) ISO 8571 : International Standards Organisation
- g) ISO 8572 : International Standards Organisation
- h) ISO/IEC 8824:1990: Specification of ASN.1
- i) ISO/IEC 8824-1/2/3/4:1998 Various enhancements to ASN.1 specification
- j) ISO/IEC 8825:1990 Specification of Basic Encoding Rules for ASN.1
- k) ISO/IEC 8825-1:1998 Specification of Basic Encoding Rules (BER) Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). Informative references
- l) I.410 : International Telecommunications Union.
- m) V.24/V.28 : International Telecommunications Union
- n) V.35 : International Telecommunications Union
- o) X.21 : International Telecommunications Union
- p) X.21 bis : International Telecommunications Union
- q) RFC 768: Internet Engineering Task Force
- r) RFC 791 : Internet Engineering Task Force
- s) RFC 793: Internet Engineering Task Force
- t) RFC 821 : Internet Engineering Task Force
- u) RFC 854 : Internet Engineering Task Force
- v) RFC 959 : Internet Engineering Task Force
- w) RFC 1155: Internet Engineering Task Force
- x) RFC 1157 : Internet Engineering Task Force
- y) RFC 1212: Internet Engineering Task Force
- z) RFC 1945: Internet Engineering Task Force
- aa) CORBA: OMG
- bb) SOAP: W3C
- cc) XML: W3C (see also OASIS)
- dd) UDDI: OASIS

A.2 Further information may be found on the following websites:

- a) UTMC: [www.utmc.gov.uk](http://www.utmc.gov.uk)
- b) DfT: [www.dft.gov.uk](http://www.dft.gov.uk)
- c) Highways Agency: [www.highways.gov.uk](http://www.highways.gov.uk)
- d) ITU: [www.itu.int](http://www.itu.int)
- e) ISO: [www.iso.ch](http://www.iso.ch)
- f) CEN: [www.cenorm.be](http://www.cenorm.be)
- g) IETF: [www.ietf.org](http://www.ietf.org)
- h) OMG: [www.omg.org](http://www.omg.org)
- i) OASIS: [www.oasis-open.org](http://www.oasis-open.org)
- j) W3C: [www.w3.org](http://www.w3.org)
- k) e-GIF: [www.e-Envoy.gov.uk](http://www.e-Envoy.gov.uk)
- l) TIH: [www.tih.org.uk](http://www.tih.org.uk)
- m) RTIG: [www.rtig.org.uk](http://www.rtig.org.uk)
- n) TPEG: [www.ebu.ch/en/technical/projects/b\\_tpeg.php](http://www.ebu.ch/en/technical/projects/b_tpeg.php)

## **B Management authority for this document (Informative)**

### **B.1 Formal authority**

B.1.1 The UK Department for Transport is the management authority for the UTMC Technical Specification.

B.1.2 The contact address for formal matters is:

Traffic Management Division  
Department for Transport  
Great Minster House  
76 Marsham Street  
London  
SW1P 4DR  
UK

### **B.2 Day to day management**

B.2.1 DfT has delegated the task of managing the Technical Specification to the UTMC Development Group (UDG), a cooperative grouping of local authorities and system suppliers. The UDG Specifications and Standards Group manages the document on a day to day basis. The UDG secretariat is the point of contact from which up to date copies of the Technical Specification can be obtained.

B.2.2 The contact address for day to day matters is currently:

UDG Technical Secretariat  
c/o Centaur Consulting Limited  
Surrey Technology Centre  
Surrey Research Park  
Guildford  
Surrey GU2 7YG  
UK

Tel: +44 (0) 1483 688270  
Fax: +44 (0) 1483 688271  
E-mail: [utmc@centaurconsulting.co.uk](mailto:utmc@centaurconsulting.co.uk)

B.2.3 Any changes to this contact address will be published on the UTMC website.

## C Schedule of registered objects (Informative)

### C.1 Registered Objects

C.1.1 The table below shows the current list of approved entries in the UTMC Objects Registry, including version number and date of the current release. This table only includes entries that have been specified in detail (ie have got at least as far as Submitted status).

#### *Data Objects (see Annex D)*

Reference	Name	Status	Current version/TS004 entry
UC/001	Access Control Data Object	Submitted	30/3/05
UC/002	Accident Data Object	Submitted	30/3/05
UC/003	Air Quality Data Object	Submitted	30/3/05
UC/004	Car Park Data Object	Submitted	30/3/05
UC/005	CCTV Data Object	Submitted	30/3/05
UC/006	Common Support Types Data Objects <i>Quality Support Object</i> <i>Device History Support Object</i> <i>Command Support Object</i>	Submitted	30/3/05
UC/007	Common Subsystem Support Object	Submitted	30/3/05
UC/008	Common TypeID Support Object	Submitted	30/3/05
UC/009	DayType Support Object	Submitted	30/3/05
UC/010	Detector Data Object	Submitted	30/3/05
UC/011	Event Data Object	Submitted	30/3/05
UC/012	Fault Support Objects <i>Fault Support Object</i> <i>Acknowledgement State Support Object</i> <i>Fault Type Support Object</i>	Submitted	30/3/05
UC/013	Global Support Object	Submitted	30/3/05
UC/014	Incident Data Object	Submitted	30/3/05
UC/015	Meteorological Data Object	Submitted	30/3/05
UC/016	Network Support Objects <i>Network Node Support Object</i> <i>Network Link Support Object</i> <i>Network Turn Support Object</i> <i>Network Geometry Support Object</i> <i>Network Path Support Object</i> <i>Network Zone Support Object</i>	Submitted	30/3/05
UC/017	Prediction Data Object	Submitted	30/3/05
UC/018	Profile Data Object	Submitted	30/3/05
UC/019	Roadworks Data Object	Submitted	30/3/05
UC/020	Traffic Signal Data Object	Submitted	30/3/05
UC/021	Transport Link Data Object	Submitted	30/3/05
UC/022	Transport Route Data Object	Submitted	30/3/05

## C Schedule of registered objects (Informative)

Reference	Name	Status	Current version/TS004 entry
UC/023	VMS Data Object VMS Messages Support Object VMS Message List Support Object VMS Car Park List Support Object	Submitted	30/3/05

### **MIBs (see Annex E)**

Reference	Name	Status	Current version	TS004 entry
UM/001	UTMC Header MIB	Submitted	1.05	30/3/05
UM/002	Air Quality Monitor MIB	Submitted	2.01	30/3/05
UM/003	VMS MIB	Submitted	3.01	30/3/05
UM/004	Simple UTC MIB	Submitted	0.2	30/3/05
UM/005	UTC MIB	Submitted	1.12	30/3/05
UM/006	Car Park Monitor MIB	Submitted	2.01	30/3/05
UM/007	Traffic Counter MIB	Submitted	4.01	30/3/05

C.1.2 A number of MIBs have associated guidance documents, which are available separately.

### **IDL scripts (see Annex F)**

Reference	Name	Status	Current version/TS004 entry
UI/001	BCD.idl	Submitted	30/3/05
UI/002	MJD.idl	Submitted	30/3/05
UI/003	B-Query.idl	Submitted	30/3/05
UI/004	B-SessionManagement.idl	Submitted	30/3/05
UI/005	B-Subscriptions.idl	Submitted	30/3/05
UI/006	B-TabularResults.idl	Submitted	30/3/05
UI/007	B-Utility.idl	Submitted	30/3/05

### **Other objects (see Annex G)**

C.1.3 There are currently no other registered Objects.

## **D Definitions of registered Data Objects (Normative)**

- D.1 Data Objects for use in UTMC systems are described in the spreadsheet “TS004 Annex D Mar05” that accompanies this document.
- D.2 These standardised Data Objects should be used wherever possible to structure the data passed across an open interface within a UTMC system.
- D.3 There is no obligation to use these Objects within applications, within a Common Database, or across a link which has been defined as non-open in the governing system architecture, so long as the relevant modules are capable of receiving and providing data in these structures.

## E Definitions of registered MIBs (Normative)

### E.1 Introduction

- E.1.1 MIBs for use in UTMC systems are described in the spreadsheet "TS004 Annex E Mar05" that accompanies this document.
- E.1.2 These standardised MIBs should be used wherever possible to manage the data passed across an SNMP communications link within a UTMC system.

### E.2 UM/003, UTMC header MIB

UTMC-Header-MIB DEFINITIONS ::= BEGIN

```
--      Y1-01017.txt
--      Revision: 1.05
--      Product No:      UTMC Header MIB
--      Date:            22/2/2005
--      Written: Robin Jefferson

--      Revision History
--      V1.00   13/5/2002   First Issue                RLJ
--      V1.01   24/5/2002   Car Parks sub-branch added  RLJ
--      V1.02   15/12/2004  Rising Bollard sub-branch added  RLJ
--      V1.03   18/2/2005   Add Common definitions         RLJ
--      V1.05   22/2/2005   Modify True/False, Add Time format  RLJ

--      City of York Council
--      9 St Leonard's Place
--      York
--      YO1 7ET
--      Tel +44 1904 551372
--      Fax +44 1904 551412

--      Maintained by
--      Integrated Design Techniques Ltd
--      Endurance House
--      Seventh Avenue
--      Team Valley
--      Tyne & Wear
--      NE11 0EF
--      Tel +44 191 491 0800
--      Fax +44 191 491 0799
--      email:  robin@idtuk.com

--      This module provides definitions and registration points for
--      City of York Council's UTMC compliant outstations

--      City of York Council reserve the right to make changes in this specification
--      and other information contained in this document without
--      prior notice. In no event shall City of York Council be liable for any
--      incidental, indirect, special or consequential damages arising out of, or
```

```
-- related to the use of this document or the information contained in it.
-- City of York Council grant vendors and end-users a non-exclusive
-- licence to use this specification in the connection with management
-- of UTMC compliant outstations.

-- Copyright City of York Council 2002

-- Defined OIDs from RFC1155-SMI

ccitt      OBJECT IDENTIFIER ::= { 0 }
null      OBJECT IDENTIFIER ::= { ccitt 0 }

iso       OBJECT IDENTIFIER ::= { 1 }
org       OBJECT IDENTIFIER ::= { iso 3 }
dod       OBJECT IDENTIFIER ::= { org 6 }
internet  OBJECT IDENTIFIER ::= { dod 1 }
directory OBJECT IDENTIFIER ::= { internet 1 }
mgmt      OBJECT IDENTIFIER ::= { internet 2 }
experimental OBJECT IDENTIFIER ::= { internet 3 }
private   OBJECT IDENTIFIER ::= { internet 4 }
enterprises OBJECT IDENTIFIER ::= { private 1 }

-- Mod V1.03 - Add common definitions
-- DisplayString ::= OCTET STRING
-- This data type is defined to support textual information using
-- the ASCII character set. By convention, objects declared with this
-- syntax, unless otherwise specified are declared as having:
--
-- SIZE (0..255)

-- TruthValue ::= INTEGER{true (1), false (2)}
-- UTMCTime ::= DisplayString (SIZE(13))
-- This object sets or returns the current time as YYMMDDHHmmssZ. Z indicates zulu or GMT

-- CoYC UTMC OID

utmc      OBJECT IDENTIFIER ::= { enterprises 13267 }

-- UTMC sub-branches - Registration points

-- Air Quality

utmcAirQualityMonitor OBJECT IDENTIFIER ::= { utmc 1 }
utmcAirQualType1 OBJECT IDENTIFIER ::= { utmcAirQualityMonitor 1 }

-- Dial Up UTC

utmcDialUpUTC OBJECT IDENTIFIER ::= { utmc 2 }
utmcDialUpUTCType1 OBJECT IDENTIFIER ::= { utmcDialUpUTC 1 }

-- Full UTC
```

```
utmcFullUTC          OBJECT IDENTIFIER ::= { utmc 3}
utmcFullUTCType1    OBJECT IDENTIFIER ::= { utmcFullUTC 1}

-- Simple UTC

utmcSimpleUTC        OBJECT IDENTIFIER ::= { utmc 4}
utmcSimpleUTCType1  OBJECT IDENTIFIER ::= { utmcSimpleUTC 1}

-- Traffic Counter

utmcTrafficCounter   OBJECT IDENTIFIER ::= { utmc 5}
utmcTrafficCounterType1 OBJECT IDENTIFIER ::= { utmcTrafficCounter 1}

-- VMS

utmcVMS              OBJECT IDENTIFIER ::= { utmc 6}
utmcVMSType1         OBJECT IDENTIFIER ::= { utmcVMS 1}

-- Car Parks

utmcCarParks         OBJECT IDENTIFIER ::= { utmc 7}
utmcCarParksType1    OBJECT IDENTIFIER ::= { utmcCarParks 1}

-- Rising Bollard

utmcRisingBollard    OBJECT IDENTIFIER ::= { utmc 8}
utmcRisingBollardType1 OBJECT IDENTIFIER ::= { utmcRisingBollard 1}

END
```

### E.3 UM/004, Air quality monitor MIB

UTMC-AirQualityMonitor DEFINITIONS ::= BEGIN

```
--      Y1-07010.txt
--      Revision: 2.01
--      Product No:      Air Quality Monitor
--      Date:            23/2/2005
--      Written: Robin Jefferson

--      Revision History
--      V1.00  27/5/2002      Re-drafted from Learian Designs Streetbox MIB RLJ
--      V1.01  31/5/2002      Change Maximum Temperature to read-write RLJ
--      V1.02  31/5/2002      Add Fault trap RLJ
--      V1.03  14/6/2002      Change aqdPeriod to Mandatory RLJ
--      V1.04  18/6/2002      Change to trap description RLJ
--                               Add Real time clock object
--      V1.05  3/3/2003        Change CO from ppm to ppb in objects RLJ
--                               aqcThresholdCO, aqcCO
--      V1.06  23/4/2003        Align reported values to nearest SI unit RLJ
--                               Units used parts per trillion ppt,
--                               1,000th of a degree C
--                               nano-gramme per metre cubed ng/m3
--      V1.07  24/11/2003      Addition of Port numbers and return IP Address RLJ
--      V2.01  23/2/2005      Modifications following harmonisation RLJ

--      City of York Council
--      9 St Leonard's Place
--      York
--      YO1 7ET
--      Tel +44 1904 551372
--      Fax +44 1904 551412

--      Maintained by
--      Integrated Design Techniques Ltd
--      Endurance House
--      Seventh Avenue
--      Team Valley
--      Tyne & Wear
--      NE11 0EF
--      Tel +44 191 491 0800
--      Fax +44 191 491 0799
--      email: robin@idtuk.com

--      This module provides definitions and registration points for
--      City of York Council's UTMC compliant Air Quality Monitors

--      City of York Council reserve the right to make changes in this specification
--      and other information contained in this document without
```

```
-- prior notice. In no event shall City of York Council be liable for any
-- incidental, indirect, special or consequential damages arising out of, or
-- related to the use of this document or the information contained in it.
-- City of York Council grant vendors and end-users a non-exclusive
-- licence to use this specification in the connection with management
-- of UTM C compliant outstations.

-- Copyright City of York Council 2002
```

IMPORTS

```
TRAP-TYPE
    FROM RFC-1215
OBJECT-TYPE
    FROM RFC-1212
    utmc, utmcAirQualType1, UTMCTime, DisplayString
        FROM UTM C-Header-MIB;

    SMI IpAddress

-- Textual conventions
--Mod V2.01 - Moved to Header MIB
--DisplayString ::= OCTET STRING
-- This data type is defined to support textual information using
-- the ASCII character set. By convention, objects declared with this
-- syntax, unless otherwise specified are declared as having:
--
-- SIZE (0..255)

-- the path to the root

general          OBJECT IDENTIFIER ::= { utmcAirQualType1 1 }
airQualityData  OBJECT IDENTIFIER ::= { utmcAirQualType1 2 }
airQualityConfig OBJECT IDENTIFIER ::= { utmcAirQualType1 3 }
meteorologicalData OBJECT IDENTIFIER ::= { utmcAirQualType1 4 }
fault           OBJECT IDENTIFIER ::= { utmcAirQualType1 5 }
```

-----  
-- Numerical conversions  
-----

```
-- Parts per million = ppm
-- Parts per billion = ppb
-- Parts per trillion = ppt
-- 1 ppm = 1,000 ppb = 1,000,000 ppt

-- E.g. A value measured in ppm is multiplied by 1,000,000 for transmission
-- (transmitted in ppt) and divided by 1,000,000 before entering in the CDB.

-- micro-gramme per meter cubed = ug/m3
-- nano-gramme per metre cubed = ng/m3
-- 1 ug/m3 = 1,000 ng/m3
```

-----  
 -- General and Identification objects  
 -----

--Mod V2.01 - Return threshold value in traps

aqmCOAlarm TRAP-TYPE  
 ENTERPRISE utmc  
 VARIABLES { aqdCO, aqcThresholdCO }  
 DESCRIPTION  
 "This trap indicates that the CO threshold level has been exceeded.  
 The current value of CO is returned. A single trap is sent when the  
 threshold is exceeded and again when the level falls below the threshold."  
 ::= 0

aqmNOAlarm TRAP-TYPE  
 ENTERPRISE utmc  
 VARIABLES { aqdNO, aqcThresholdNO }  
 DESCRIPTION  
 "This trap indicates that the NO threshold level has been exceeded.  
 The current value of NO is returned. A single trap is sent when the  
 threshold is exceeded and again when the level falls below the threshold."  
 ::= 1

aqmNOXAlarm TRAP-TYPE  
 ENTERPRISE utmc  
 VARIABLES { aqdNOX, aqcThresholdNOX }  
 DESCRIPTION  
 "This trap indicates that the NOX threshold level has been exceeded.  
 The current value of NOX is returned. A single trap is sent when the  
 threshold is exceeded and again when the level falls below the threshold."  
 ::= 2

aqmNO2Alarm TRAP-TYPE  
 ENTERPRISE utmc  
 VARIABLES { aqdNO2, aqcThresholdNO2 }  
 DESCRIPTION  
 "This trap indicates that the NO2 threshold level has been exceeded.  
 The current value of NO2 is returned. A single trap is sent when the  
 threshold is exceeded and again when the level falls below the threshold."  
 ::= 3

aqmOZONEAlarm TRAP-TYPE  
 ENTERPRISE utmc  
 VARIABLES { aqdOZONE, aqcThresholdOZONE }  
 DESCRIPTION  
 "This trap indicates that the Ozone threshold level has been exceeded.  
 The current value of Ozone is returned. A single trap is sent when the  
 threshold is exceeded and again when the level falls below the threshold."  
 ::= 4

aqmPAHAlarm TRAP-TYPE

```

ENTERPRISE utmc
VARIABLES      { aqdPAH, aqcThresholdPAH }
DESCRIPTION
    "This trap indicates that the PAH threshold level has been exceeded.
    The current value of PAH is returned. A single trap is sent when the
    threshold is exceeded and again when the level falls below the threshold."
 ::= 5
  
```

```

aqmSO2Alarm TRAP-TYPE
ENTERPRISE utmc
VARIABLES      { aqdSO2, aqcThresholdSO2 }
DESCRIPTION
    "This trap indicates that the SO2 threshold level has been exceeded.
    The current value of SO2 is returned. A single trap is sent when the
    threshold is exceeded and again when the level falls below the threshold."
 ::= 6
  
```

```

aqmPM10Alarm TRAP-TYPE
ENTERPRISE utmc
VARIABLES      { aqdPM10, aqcThresholdPM10 }
DESCRIPTION
    "This trap indicates that the PM10 threshold level has been exceeded.
    The current value of PM10 is returned. A single trap is sent when the
    threshold is exceeded and again when the level falls below the threshold."
 ::= 7
  
```

```

aqmBTXAlarm TRAP-TYPE
ENTERPRISE utmc
VARIABLES      { aqdBTX, aqcThresholdBTX }
DESCRIPTION
    "This trap indicates that the BTX threshold level has been exceeded.
    The current value of BTX is returned. A single trap is sent when the
    threshold is exceeded and again when the level falls below the threshold."
 ::= 8
  
```

```

faultAlarm      TRAP-TYPE
ENTERPRISE utmc
VARIABLES      { faultNo, faultID, faultSeverity, faultDate, faultRTC }
DESCRIPTION
    "This trap indicates that a fault has occurred.
    The last recorded fault is returned."
 ::= 9
  
```

-----  
 -- General and Identification objects  
 -----

--Mod V2.01 - Deprecate these objects as they are duplicated in the common database

```

genSystemCodeNumber OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-write
  
```

STATUS deprecated  
 DESCRIPTION  
     "Unique identifier for the device"  
     ::= { general 1 }

genName OBJECT-TYPE  
 SYNTAX OCTET STRING  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Human readable name for the device"  
     ::= { general 2 }

genLongDescription OBJECT-TYPE  
 SYNTAX DisplayString  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Long description of the device"  
     ::= { general 3 }

genNorthing OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Location of device in OS grid coordinates"  
     ::= { general 4 }

genEasting OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Location of device in OS grid coordinates"  
     ::= { general 5 }

genLinkReference OBJECT-TYPE  
 SYNTAX OCTET STRING  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Reference to the link on which the device resides"  
     ::= { general 6 }

genLinkDistance OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
     "Distance between point and start of link"  
     ::= { general 7 }

--Mod V2.01 - Deprecate and use genRTC

genTime OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS deprecated  
DESCRIPTION

"Sets or returns the current time (in seconds since 1st January 1970 00:00:00)."

::= { general 8 }

genIPAddress OBJECT-TYPE  
SYNTAX IpAddress  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

" This object hold the IP Address to which traps are returned. If the object is invalid or 0.0.0.0 then traps are returned to the IP Address of the manager which last made a Set or Get request"

::= { general 9 }

genPort OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

" This object hold the Port number to which traps are returned. If the object is invalid or 0 then traps are returned to the local Port of the manager which last made a Set or Get request"

::= { general 10 }

--Mod V2.01 - Add RTC

genRTC OBJECT-TYPE  
SYNTAX UTMCTime  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"Sets or returns the current time."

::= { general 11 }

--Mod V2.01 - Add Software version

genSoftwareVer OBJECT-TYPE  
SYNTAX DisplayString  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"Returns the software version as Vmajor.minor."

::= { general 12 }

-----  
-- Air Quality Data & Measurements  
-----

--Mod V2.01 Deprecate and replace with aqdStartRTC

aqdStartTime OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-write  
STATUS deprecated  
DESCRIPTION

"Sets or returns the Start time to which dynamic data (air quality & Meteorological) relates.

(Time in seconds since 1st January 1970 00:00:00)"

::= { airQualityData 1 }

aqdPeriod OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"Sets or returns the period in minutes over which the data was collected.

Some AQMs are fixed to 15 minutes. Setting a time period which is not

supported

causes the value to be unchanged and a badValue error is raised."

::= { airQualityData 2 }

aqdCO OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"CO concentration in ppt"

::= { airQualityData 3 }

aqdNO OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"NO concentration in ppt"

::= { airQualityData 4 }

aqdNOX OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"NOX concentration in ppt"

::= { airQualityData 5 }

aqdNO2 OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"NO2 concentration in ppt"  
 ::= { airQualityData 6 }

aqdOZONE OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"Ozone concentration in ppt"  
 ::= { airQualityData 7 }

aqdPAH OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"PAH concentration in ppt"  
 ::= { airQualityData 8 }

aqdSO2 OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"SO2 concentration in ppt"  
 ::= { airQualityData 9 }

aqdPM10 OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"PM10 concentration in ng/m3"  
 ::= { airQualityData 10 }

aqdBTX OBJECT-TYPE

SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"BTX concentration in ppt"  
 ::= { airQualityData 11 }

aqdSummary OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (36))  
 ACCESS read-only  
 STATUS mandatory      DESCRIPTION

"OCTET-STRING returning all measurements as fixed field 32 bit Integers

(Little Endian)

CO      INTEGER  
 NO      INTEGER  
 NOX     INTEGER

NO2 INTEGER  
OZONE INTEGER  
PAH INTEGER  
SO2 INTEGER  
PM10 INTEGER  
BTX INTEGER

NULL values returned for unused values"  
::= { airQualityData 12 }

--Mod V2.01 - Add RTC  
aqdStartRTC OBJECT-TYPE  
SYNTAX UTMCTime  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"Sets or returns the current time"  
::= { airQualityData 13 }

-----  
-- Air Quality thresholds  
-----

aqcThresholdCO OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION  
"CO concentration threshold in ppt"  
::= { airQualityConfig 1 }

aqcThresholdNO OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION  
"NO concentration threshold in ppt"  
::= { airQualityConfig 2 }

aqcThresholdNOX OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION  
"NOX concentration threshold in ppt"  
::= { airQualityConfig 3 }

aqcThresholdNO2 OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION

"NO2 concentration threshold in ppt"  
 ::= { airQualityConfig 4 }

aqcThresholdOZONE OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "Ozone concentration threshold in ppt"  
 ::= { airQualityConfig 5 }

aqcThresholdPAH OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "PAH concentration threshold in ppt"  
 ::= { airQualityConfig 6 }

aqcThresholdSO2 OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "SO2 concentration threshold in ppt"  
 ::= { airQualityConfig 7 }

aqcThresholdPM10 OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "PM10 concentration threshold in ng/m3"  
 ::= { airQualityConfig 8 }

aqcThresholdBTX OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "BTX concentration threshold in ppt"  
 ::= { airQualityConfig 9 }

--Mod V2.01 - Add a trapTrigger object

aqcTrapTrigger OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "This object sets which state to trigger a trap. A trap will trigger only once on each entry to trap configured state"

```

b0 (1) - CO
b1 (2) - NO
b2 (4) - NOX
b3 (8) - NO2
b4 (16)- OZONE
b5 (32)- PAH
b6 (64)- SO2
b7 (128)- PM10
b8 (256)- BTX
b9 (512)- Fault"
 ::= { airQualityConfig 10 }

```

---

-- Meteorological Data

---

mldRoadCondition OBJECT-TYPE

```

SYNTAX INTEGER {
    dry(1),
    wet(2),
    icy(3)
}

```

ACCESS read-only

STATUS optional

DESCRIPTION

"Indication of road condition"

::= { meteorologicalData 1 }

mldRoadTemp OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS optional

DESCRIPTION

"Current road temperature (1,000ths of a degree C)

E.g. 23.2C is reported as 23,200."

::= { meteorologicalData 2 }

mldVisibilityType OBJECT-TYPE

```

SYNTAX INTEGER {
    good-vis(1),
    moderate-vis(2),
    poor-vis(3),
    haze(4),
    mist(5),
    slight-fog(6),
    fog(7),
    dense-fog(8)
}

```

ACCESS read-only

STATUS optional

DESCRIPTION

"Indication of visibility problems"

Good Visibility	- >10km
Moderate Visibility	- 4 - 10km
Poor Visibility	- 2 - 4km
Haze	- 1 - 2km
Mist	- 1 - 2km
Slight Fog	- 180m - 1km
Fog	- 45 - 180m
Dense Fog	- <45m"

::= { meteorologicalData 3 }

mldVisibilityDist OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
     "Visibility distance in meters"  
 ::= { meteorologicalData 4 }

mldPrecipitationType OBJECT-TYPE  
 SYNTAX INTEGER {  
     none(1),  
     drizzle(2),  
     rain(3),  
     hail(4),  
     snow(5)  
 }  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
     "Type of precipitation"  
 ::= { meteorologicalData 5 }

mldPrecipitationIntensity OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
     "Indication of precipitation intensity (in micro-metres (um) over a 15min  
 period)  
     E.g. 2.3mm of percipitation in a 15 minute period is transmitted as 2,300."  
 ::= { meteorologicalData 6 }

mldAirTemp OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
     "Current air temperature (1,000ths of a degree C)"  
 ::= { meteorologicalData 7 }

mldMinTemp OBJECT-TYPE  
 SYNTAX INTEGER

ACCESS read-write  
STATUS optional  
DESCRIPTION

"Minimum recorded temperature (1,000ths of a degree C).  
Reset by writing in a default value (+100,000)."  
::= { meteorologicalData 8 }

mldMaxTemp OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION

"Maximum recorded temperature (1,000ths of a degree C).  
Reset by writing in a default value (-100,000)."  
::= { meteorologicalData 9 }

mldHumidity OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"Current relative humidity level (1,000ths of a %RH)  
E.g. 60.4% is reported as 60,400."  
::= { meteorologicalData 10 }

mldPressure OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"Atmospheric pressure (1,000ths of a mB (hPa))  
E.g. 1013.2mB is reported as 1013,200."  
::= { meteorologicalData 11 }

mldWindSpeed OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS optional  
DESCRIPTION

"Average wind speed (1,000ths of a metre/second)"  
::= { meteorologicalData 12 }

mldMaxWindSpeed OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-write  
STATUS optional  
DESCRIPTION

"Maximum recorded wind speed (1000ths of a metre/second)  
Reset by writing in a default value (0)."  
::= { meteorologicalData 13 }

mldWindDirection OBJECT-TYPE

```

SYNTAX INTEGER (0..360)
ACCESS read-only
STATUS optional
DESCRIPTION
    "Degrees from device north"
    ::= { meteorologicalData 14 }

```

-----  
-- Faults  
-----

```

faultTable      OBJECT-TYPE
    SYNTAX SEQUENCE OF FaultEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Provides a table of instances of fault data."
    ::= { fault 1 }

```

```

faultEntry      OBJECT-TYPE
    SYNTAX FaultEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of faults"
    INDEX { faultNo }
    ::= { faultTable 1 }

```

```

FaultEntry ::=
    SEQUENCE {
        faultNo      INTEGER,
        faultID      INTEGER,
        faultType     INTEGER,
        faultSeverity INTEGER,
        faultDate     INTEGER,
        faultRTC      UTMCTime
    }

```

```

faultNo      OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Returns the index for a fault entry"
    ::= { faultEntry 1 }

```

```

faultID      OBJECT-TYPE

```

```

SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Returns the identifier for the fault type"
 ::= { faultEntry 2}

--Mod V2.01 - Deprecate
faultType OBJECT-TYPE
    SYNTAX INTEGER {
        equipment(1),
        comms(2)
    }
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "Returns the sub-type for the fault type"
    ::= { faultEntry 3}

--Mod V2.01 - Deprecate
faultSeverity OBJECT-TYPE
    SYNTAX INTEGER (0..1000)
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "Returns the fault severity"
    ::= { faultEntry 4}

--Deprecate and replace with faultRTC
faultDate OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "Returns the time & date for the fault in seconds since 1st Jan 1970."
    ::= { faultEntry 5}

faultRTC OBJECT-TYPE
    SYNTAX UTMCTime
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Returns the time & date for the fault."
    ::= { faultEntry 6}

END

```

E.4 UM/006, VMS MIB

--IDENTIFICATION

-- Module : VMSUTMC.mib
-- Version : V3.01
-- Author : A Kipling
-- Date : 25/01/2005

-- Function:
-- For the control and management of Variable Message Signs via the SNMP Protocol

-- VMS object definitions MIB

-- Variable Message Signs Limited
-- Unit 1,
-- Monkton Business Park North,
-- Mill Lane,
-- Hebburn,
-- Tyne & Wear
-- NE31 2JZ,
-- United Kingdom

-- Modified 06/06/2002 to include CoYC Header File - ALK
-- Modified 29/10/2002 - updated description of objects - ALK
-- Modified 29/10/2002 - Changed ACCESS on msgTime and statusTime to READ only
-- Modified 29/10/2002 - added vmsSetTime and vmsPort objects.
-- Modified 31/07/2003 - added vmsCommsCheckStatus node
-- Modified 31/07/2003 - added
vmsCommsCheckcoms,vmsCheckTimer,vmsBlankOnFault,vmsTimeOut,trapExtcomms
-- Modified 25/01/2005 - TruthTable Definition has been removed and added to the header.mib
-- Modified 25/01/2005 - Updated description on vmsMibSoftwareVersion, vmsMaxHeight,
vmsMaxWidth, vmsMaxFontSpacing
-- Modified 25/01/2005 - Updated description on vmsMaxFontHeight, vmsMaxFontWidth,
vmsMinHeight, vmsMinWidth
-- Modified 25/01/2005 - Updated description on vmsMinFontSpacing, vmsMinFontHeight,
vmsMinFontWidth
-- Modified 25/01/2005 - Updated description on signID, vmsPassword, signType, vmsConfigTime,
vmsHeight, vmsWidth
-- Modified 25/01/2005 - Updated description on vmsFontSpacing, vmsFontHeight, vmsFontWidth
-- Modified 25/01/2005 - Updated description on vmsReturnIpAddress, vmsLogIn, vmsSetTime,
vmsPort, displayText
-- Modified 25/01/2005 - Updated description on msgTime, vmsLuminanceOverride, vmsLuminance,
statusTime
-- Modified 25/01/2005 - faultDescription, numberFaults objects added to the vmsFaultStatus node.
-- Modified 25/01/2005 - Updated description on vmsCommsCheck, vmsCheckTimer,
-- Modified 25/01/2005 - faultChange TRAP added.

=====

--VARIABLE MESSAGE SIGNS (VMS) OBJECTS

VMS DEFINITIONS ::= BEGIN

IMPORTS

TRAP-TYPE	FROM RFC-1215
OBJECT-TYPE	FROM RFC-1212
IpAddress, enterprises	FROM RFC1155-SMI
utmc, utmcVMS, utmcVMSType1, TruthTable	FROM Header-MIB;

--For the purpose of this section, the following OBJECT IDENTIFIERS are used:  
 --the node location is: private/enterprises/utmc/utmcVMS/utmcVMSType1

sysInfo OBJECT IDENTIFIER ::= { utmcVMSType1 1 }

--This node is used to define the limits in which the VMS has to operate within.

vmsMibSoftwareVersion OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE (0..255))  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "The current MIB version being used by the vms. Version 3.01 will be stored as 'v3.01'"  
 ::= { sysInfo 1 }

vmsMaxHeight OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "This object holds the maximum number of rows the VMS sign can display.  
 if the object is not used a default value of 0 (zero) should be entered."  
 ::= { sysInfo 2 }

vmsMaxWidth OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "This object holds the maximum number of characters the VMS sign can display per  
 line.  
 if the object is not used a default value of 0 (zero) should be entered."  
 ::= { sysInfo 3 }

vmsMaxFontSpacing OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "This object holds the maximum value of the font spacing (in pixels) allowed on the  
 VMS sign.  
 if the object is not used a default value of 0 (zero) should be entered."  
 ::= { sysInfo 4 }

vmsMaxFontHeight OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the maximum font height (in pixels) the VMS sign can display.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 5}

vmsMaxFontWidth OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the maximum font width (in pixels) the VMS sign can display.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 6}

vmsLanternsPresent OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Does the sign have 'Flashing Lanterns?', 1=True, 2=False"  
::= {sysInfo 7}

vmsMinHeight OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the minimum number of rows the VMS sign can support.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 8}

vmsMinWidth OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the minimum number of characters the VMS sign can support per  
row.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 9}

vmsMinFontSpacing OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the minimum value of the font spacing (in pixels) allowed on the  
VMS sign.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 10}

vmsMinFontHeight OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only

STATUS mandatory  
DESCRIPTION "This object holds the minimum font height (in pixels) the VMS sign can display.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysInfo 11}

vmsMinFontWidth OBJECT-TYPE  
SYNTAX INTEGER (0..100)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "This object holds the minimum font width (in pixels) the VMS sign can display.  
if the object is not used a default value of 0 (zero) should be entered."  
::= {sysinfo 12}

sysConfig OBJECT IDENTIFIER ::= { utmcVMSType1 2 }

--This node is used to give the current settings of the VMS.

signID OBJECT-TYPE  
SYNTAX INTEGER (0..255)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "the Unique ID of the VMS. if the object is not used a default value of 0 (zero) should  
be entered."  
::= {sysConfig 1}

vmsPassword OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE (0..50))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "The current Password must be given to allow the sign to be used. A NULL string will  
be used to indicate that no password is required to log onto the system. The use of 'logoff' is to  
be prevented as this is used to log the user off from the system."  
::= {sysConfig 2}

signType OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE(0..255))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "Textual Description of the sign type currently been used. If this object is not used a  
default NULL string will be entered"  
::= {sysConfig 3}

vmsLanterns OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "Indicates if any lanterns present are available for use on this VMS"  
::= {sysConfig 4}

vmsConfigTime OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE (11))  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Displays the time of the current config settings. The Format is YYMMDDHHmmZ where Z represents GMT Timezone. If this object is not used a default value of '0000000000Z' is to be entered."  
 ::= {sysConfig 5}

vmsHeight OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Indicates the maximum number of rows available for message display (eg 4). If this object is not used a default value of 0 (zero) is to be entered."  
 ::= {sysConfig 6}

vmsWidth OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Indicates the maximum number of characters per line (eg 15). If this object is not used a default value of 0 (zero) is to be entered."  
 ::= {sysConfig 7}

vmsFontSpacing OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Number of pixels between characters (eg 2). If this object is not used a default value of 0 (zero) is to be entered."  
 ::= {sysConfig 8}

vmsFontHeight OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "The height of the vms font in pixels (eg 5). If this object is not used a default value of 0 (zero) is to be entered."  
 ::= {sysConfig 9}

vmsFontWidth OBJECT-TYPE  
 SYNTAX INTEGER(0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "The width of the vms font in pixels (eg 7). If this object is not used a default value of 0 (zero) is to be entered."  
 ::= {sysConfig 10}

vmsReturnIpAddress OBJECT-TYPE

SYNTAX IpAddress  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "This object holds the IP Address to which traps are returned. If the object is invalid or 0.0.0.0 (default value) then traps are returned to the IP Address of the manager which last made a SET or GET request"  
 ::= {sysConfig 11}

vmsLogIn OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(0..50))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "This object is written to in order to log onto the vms, the value written into here is compared the vmsPassword object. A value of 'logoff' is used to log the user off. The default value for this object is a NULL string. If access to any of the MIB objects does not occur within a 2 minute period any active user will be automatically logged off. All Passwords are case sensitive."  
 ::= {sysConfig 12}

vmsSetTime OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(11))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "This object is used to write the current system into, it allows the VMS internal clock to be update with this system time Format is YYMMDDHHmmZ where Z represents GMT Timezone. The default value of this object will be '0000000000Z'. When the time has been updated this object should return to its default value."  
 ::= {sysConfig 13}

vmsPort OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "This object holds the Port number to which traps are returned. If the object is 0 (zero) then traps are returned to the local port of the manager which last made a SET or GET request. The default value for this object will be 0 (zero)."  
 ::= {sysConfig 14}

vmsDisplayConfig OBJECT IDENTIFIER ::= {utmcVMSType1 3}

-- This Node is used to group all the objects for the VMS sign displayed messages.

messageTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF MessageTableEntry

ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION "This table holds the currently displayed messages"  
 ::= {vmsDisplayConfig 1}

messageTableEntry OBJECT-TYPE  
 SYNTAX MessageTableEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION "parameters of the Message List Table"  
 INDEX {messageLineID}  
 ::= {messageTable 1}

MessageTableEntry ::= SEQUENCE {  
     messageLineID INTEGER,  
     displayText OCTET STRING}

messageLineID OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Indicates the line number of the Message to display"  
 ::= {messageTableEntry 1}

displayText OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(0..100))  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "The contents of the line to be displayed. If a new display request is received by the VMS (and it is valid) the previous message is to be cleared from the table and the the VMS display will be updated accordingly"  
 ::= {messageTableEntry 2}

lanternsOnOff OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Indicates if the lanterns are turned On or Off for the currently displayed message"  
 ::= {vmsDisplayConfig 2}

msgTime OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(11))  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Time at which current displayed message was set. The Format is YYMMDDHHmmZ where Z represents GMT Timezone. The default value for this object is '0000000000Z'. "  
 ::= {vmsDisplayConfig 3}

vmsLuminanceOverride OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-write

STATUS mandatory  
 DESCRIPTION "This is set to 'True' if the luminance level is to be set by the operator and not by the sign. This object MUST be set to 'True' in a previous packet before you can update the vmsLuminance object."  
 ::= {vmsDisplayConfig 4}

vmsLuminance OBJECT-TYPE  
 SYNTAX INTEGER(0..15)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "Indicates the current luminance level of the vms, 0 (zero) is the lowest setting, 15 is the highest setting. vmsLuminanceOverride MUST be set to 'True' in an earlier SNMP packet before this object will accept updates. When the vmsLuminanceOverride object is set to 'True' this object should be updated to hold the default of 7 (the midway point in the luminance levels)."  
 ::= {vmsDisplayConfig 5}

vmsFaultStatus OBJECT IDENTIFIER ::= {utmcVMSType1 4}

-- Holds all the current vms faults to be reported back to the instation

faultStatus OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Indicates if the vms currently has a fault present"  
 ::= {vmsFaultStatus 1}

statusTime OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(11))  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Time at which status information was last requested the Format is YYMMDDHHmmZ where Z represents GMT Timezone. If this object is not used a default value of '0000000000Z' is to be entered."  
 ::= {vmsFaultStatus 2}

internalCommsStatus OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Indicates an internal comms failure within the VMS"  
 ::= {vmsFaultStatus 3}

messageFail OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-only

STATUS mandatory  
DESCRIPTION "Indicates message fail/watchdog reset error"  
::= {vmsFaultStatus 4}

ledFailNonCritical OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates a single led failure in the vms display modules"  
::= {vmsFaultStatus 5}

ledFailCritical OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates multiple led failures on the vms display modules"  
::= {vmsFaultStatus 6}

heaterFail OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates a heater fail within the vms unit"  
::= {vmsFaultStatus 7}

watchDogReset OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates a watchdog reset on the vms"  
::= {vmsFaultStatus 8}

overTemperature OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates an overtemperature in the vms enclosure"  
::= {vmsFaultStatus 9}

luminanceFail OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates a luminance fail in the vms"  
::= {vmsFaultStatus 10}

lanternFail OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Indicates a lantern failure on the vms"  
::= {vmsFaultStatus 11}

invalidSignAddress OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "This object is set to 1 (true) if a received signID value is greater than 255d.  
The object will be cleared automatically when a valid signID is received"

::= {vmsFaultStatus 12}

configError OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "This will object is set to 1 (true) if an invalid config is requested to be set"

::= {vmsFaultStatus 13}

powerFail OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "This object is set to 1 (true) if a power fail is detected on the VMS"

::= {vmsFaultStatus 14}

noConfigFile OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "Object set to 1 (true) if the configuration file cannot be located. Will  
only be used during startup of VMS and will not be cleared during  
normal operation. This has no use if the config file method of loading parameters  
is not used."

::= {vmsFaultStatus 15}

noSysInfoFile OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "Object set to 1 (true) if the System Info file cannot be located. Will  
only be used during startup of VMS and will not be cleared during normal  
operation. This has no use if the sysinfo file method of loading  
parameters is not used."

::= {vmsFaultStatus 16}

noSignID OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only

STATUS mandatory

DESCRIPTION "This object is to be used with the signID object."

::= {vmsFaultStatus 17}

vmsExternalCommsFault OBJECT-TYPE

SYNTAX TruthTable

ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "Indicates a comms failure to the Instation. Once set the vmsCommsCheckStatus node is disabled.  
                   Comms can only be re-instated by the Instation or a local connection."  
 ::= {vmsFaultStatus 18}

faultDescription OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(0..255))  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION "A maunfacturer specific text string used to supply a 'user-friendly' description of any faults  
                   present on the VMS."  
 ::= {vmsFaultStatus 19}

numberFaults OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION "The total number of faults present on the VMS. This is used to generate faultChange TRAP.  
                   whenever value of this objects changes a the faultChange TRAP will be raised. If this  
 object  
                   is not used a default value of 0 (zero) will be returned (this will also disable the  
 trapFaultChange TRAP)."  
 ::= {vmsFaultStatus 20}

vmsCommsCheckStatus OBJECT IDENTIFIER ::= {utmcVMSType1 5}

-- This Node is used to define the rules for the checking the external comms link to the Instation.

vmsCommsCheck OBJECT-TYPE  
 SYNTAX TruthTable  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION "If the object is set to true, the VMS will send out a extComms TRAP every 'checktimer' minutes, The  
                   Instation is expected to reply to the VMS after the extComms TRAP has been raised. Every  
 time a valid  
                   message (correctly access any MIB object) is recieved from the Instation the timer is re-set.  
 If no  
                   response is recieved to the extComms TRAP it is assumed that the comms to Instation has  
 failed, a maximum  
                   of 5 attempts to conact the in-station should be made with a delay of 1 minute between  
 TRAPS"  
 ::= {vmsCommsCheckStatus 1}

vmsCheckTimer OBJECT-TYPE  
SYNTAX INTEGER (0..1440)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "The time period for checking the external comms to the Instatation. The time period is in minutes.

If this object is not used a default value of 0 should be returned."

::= {vmsCommsCheckStatus 2}

vmsBlankOnFault OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "If the object is set to true, the VMS will clear its display if an externalcomms fault is detected."

::= {vmsCommsCheckStatus 3}

vmsTimeOut OBJECT-TYPE  
SYNTAX TruthTable  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION "Used for the extComms TRAP. This object is set true to trigger the trap once the timer has elapsed.

Object is set back to false after the TRAP has been sent, ready for the next attempt."

::= {vmsCommsCheckStatus 4}

--Trap Definitions

trapFaults TRAP-TYPE  
ENTERPRISE vmsFaultStatus  
VARIABLES {faultStatus}  
::= 1

trapExtcomms TRAP-TYPE  
ENTERPRISE vmsCommsCheckStatus  
VARIABLES {vmsTimeOut}  
::= 2

trapFaultChange TRAP-TYPE  
ENTERPRISE vmsFaultStatus  
VARIABLES {numberFaults}  
::= 3

--

END

**E.5 UM/015, Simple UTC MIB**

```

UTMC-MIB DEFINITIONS ::= BEGIN

-- MIB VERSION 0.2
--
-- IMPORTANT PLEASE READ !!
--
-- The above MIB version will be returned within the utmcSimpleSystemMIBVersion
-- object by all SNMP agents that support this MIB for the OID
--.

IMPORTS
    OBJECT-TYPE
        FROM RFC-1212
    utmcSimpleUTC
        FROM UTMC-Header-MIB;

utmcSimpleUTCType1          OBJECT IDENTIFIER ::= { utmcSimpleUTC 1 }

-- the UTMC group

utmcSimpleManagement      OBJECT IDENTIFIER ::= { utmcSimpleUTCType1 1 }

-- This node groups all objects that support the getting and setting of OSN
-- This table contains miscellaneous variables for each outstation.

utmcSimpleSystemTable     OBJECT-TYPE
    SYNTAX      SEQUENCE OF UtmcSimpleSystem
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "General control data. Each entry contains the data
                of a particular outstation."
                ::= { utmcSimpleManagement 1 }

utmcSimpleSystem          OBJECT-TYPE
    SYNTAX      UtmcSimpleSystem
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "An entry in utmcSimpleControlTable. Contains the control data
                of a particular outstation."
    INDEX{ utmcSimpleSystemIndex }
                ::= { utmcSimpleSystemTable 1 }

UtmcSimpleSystem          ::= SEQUENCE
    {
        utmcSimpleSystemIndex          INTEGER,
        utmcSimpleSystemReset          INTEGER,
        utmcSimpleSystemSoftwareType   OCTET STRING,
        utmcSimpleSystemHardwareType   OCTET STRING,
    }

```

```

        utmcSimpleSystemHardwareID      OCTET STRING,
        utmcSimpleSystemMIBVersion      OCTET STRING
    }

utmcSimpleSystemIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the out station."
               ::= { utmcSimpleSystem 1 }

utmcSimpleSystemReset OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "When set to 1, the outstation attempts a network reset.
               When set to 2, the outstation performs a software reset."
               ::= { utmcSimpleSystem 2 }

utmcSimpleSystemSoftwareType OBJECT-TYPE
    SYNTAX      OCTET STRING
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the software type"
               ::= { utmcSimpleSystem 3 }

utmcSimpleSystemHardwareType OBJECT-TYPE
    SYNTAX      OCTET STRING
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the system Hardware platform"
               ::= { utmcSimpleSystem 4 }

utmcSimpleSystemHardwareID OBJECT-TYPE
    SYNTAX      OCTET STRING
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the Hardware ID "
               ::= { utmcSimpleSystem 5 }

utmcSimpleSystemMIBVersion OBJECT-TYPE
    SYNTAX      OCTET STRING
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the version of this MIB, shown at the top this File"
               ::= { utmcSimpleSystem 6 }

```

-- The following table contains the control/reply data for each supported  
-- outstation.

```

utmcSimpleDataTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF UtmcSimpleData

```

```

ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION "A list of control/reply data. Each entry contains the data
            of a particular outstation."
            ::= { utmcSimpleManagement 2 }

utmcSimpleData OBJECT-TYPE
SYNTAX      UtmcSimpleData
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION "An entry in utmcSimpleDataTable. Contains the control/reply data
            Of a particular outstation."
INDEX{ utmcSimpleDataIndex }
            ::= { utmcSimpleDataTable 1 }

UtmcSimpleData ::= SEQUENCE
{
    utmcSimpleDataIndex          INTEGER,
    utmcSimpleDataControlByteCount  INTEGER,
    utmcSimpleDataReplyByteCount  INTEGER,
    utmcSimpleDataControlBytes    OCTET STRING,
    utmcSimpleDataReplyBytes      OCTET STRING,
}

utmcSimpleDataIndex OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "Identifies the OSN."
            ::= { utmcSimpleData 1 }

utmcSimpleDataControlByteCount OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "The number of control bytes supported by the OSN."
            ::= { utmcSimpleData 2 }

utmcSimpleDataReplyByteCount OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "The number of reply bytes supported by the OSN."
            ::= { utmcSimpleData 3 }

utmcSimpleDataControlBytes OBJECT-TYPE
SYNTAX      OCTET STRING
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The control data for the OSN."
            ::= { utmcSimpleData 4 }

```

```
utmcSimpleDataReplyBytes OBJECT-TYPE
    SYNTAX      OCTET STRING
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "The reply data from the OSN."
               ::= { utmcSimpleData 5 }
```

```
--
-- The following table describes each bit (binary digit) of
-- utmcSimpleControlBytes for each outstation.
--
```

```
utmcSimpleControlBitDescriptionTable      OBJECT-TYPE
    SYNTAX      SEQUENCE OF UtmcSimpleControlBitDescriptionEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "A list of text descriptions. Each entry describes the meaning
               of a particular control Bit to a particular outstation."
               ::= { utmcSimpleManagement 3 }
```

```
utmcSimpleControlBitDescriptionEntry OBJECT-TYPE
    SYNTAX      UtmcSimpleControlBitDescriptionEntry
    ACCESS      not-accessible
    STATUS      mandatory
    DESCRIPTION "An entry in utmcSimpleControlBitDescriptionTable. Describes the
               meaning of a particular control bit of utmcSimpleControlBytes"
    INDEX{ utmcSimpleControlBitIndex, utmcSimpleControlBitNo }
    ::= { utmcSimpleControlBitDescriptionTable 1 }
```

```
UtmcSimpleControlBitDescriptionEntry      ::= SEQUENCE
    {
        utmcSimpleControlBitIndex          INTEGER,
        utmcSimpleControlBitNo             INTEGER,
        utmcSimpleControlBitDescription    DisplayString
    }
```

```
utmcSimpleControlBitIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION "Identifies the outstation"
    ::= { utmcSimpleControlBitDescriptionEntry 1 }
```

```
utmcSimpleControlBitNo OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-write
    STATUS      mandatory
    DESCRIPTION "The offset within the control data of the bit being described."
    ::= { utmcSimpleControlBitDescriptionEntry 2 }
```

```
utmcSimpleControlBitDescription OBJECT-TYPE
    SYNTAX      DisplayString
```

```

ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The description of a bit in the control/reply data."
           ::= { utmcSimpleControlBitDescriptionEntry 3 }

--
-- The following table describes each bit (binary digit) of utmcSimpleReplyBytes for each outstation.
--

utmcSimpleReplyBitDescriptionTable OBJECT-TYPE
SYNTAX      SEQUENCE OF UtmcSimpleReplyBitDescriptionEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION "A list of text descriptions. Each entry describes the meaning
           of a particular reply bit."
           ::= { utmcSimpleManagement 4 }

utmcSimpleReplyBitDescriptionEntry OBJECT-TYPE
SYNTAX      UtmcSimpleReplyBitDescriptionEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION "An entry in utmcSimpleReplyBitDescriptionTable. Describes the
           meaning of a particular control bit of utmcSimpleControlBytes"
INDEX{ utmcSimpleReplyBitIndex, utmcSimpleReplyBitNo }
           ::= { utmcSimpleReplyBitDescriptionTable 1 }

UtmcSimpleReplyBitDescriptionEntry ::= SEQUENCE
{
    utmcSimpleReplyBitIndex  INTEGER,
    utmcSimpleReplyBitNo    INTEGER,
    utmcSimpleReplyBitDescription  DisplayString
}

utmcSimpleReplyBitIndex OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-only
STATUS      mandatory
DESCRIPTION "Identifies the outstation"
           ::= { utmcSimpleReplyBitDescriptionEntry 1 }

utmcSimpleReplyBitNo OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The offset within the reply data of the bit being described."
           ::= { utmcSimpleReplyBitDescriptionEntry 2 }

utmcSimpleReplyBitDescription OBJECT-TYPE
SYNTAX      DisplayString
ACCESS      read-write
STATUS      mandatory
DESCRIPTION "The description of a bit in the reply data."

```

::= { utmcSimpleReplyBitDescriptionEntry 3 }

END

E.6 UM/016, UTC MIB

```

--Issue 01:12 05-Sep-02  UTMC29.MY      D.R.Tate  ++
--*****++
--          TMS/VMS PVR&D                ++
--          ++
--          TRAFFIC MANAGEMENT SYSTEM INSTATION SOFTWARE  ++
-- Copyright 2001 Peek TRAFFIC LIMITED BOREHAMWOOD ENGLAND  ++
--          ++
--*****++
--          ++
-- Filename   : UTMC29.MY                ++
--          ++
-- Subsystem  :                          ++
--          ++
-- Task       :                          ++
--          ++
-- Data Document :                        ++
--          ++
-- Module Description: This module contains the M.I.B. source  ++
--          code for the SCOOT section of the UTMC29 project  ++
--          ++
--*****++
-- Author : D.R.Tate                      ++
--          ++
-- Issue   : 01    02    03    04    05  ++
--          ++
-- Date    :12-Sep-01  xx-xxx-xx  xx-xxx-xx  xx-xxx-xx  xx-xxx-xx  ++
--          ++
--*****++
-- History                                     ++
--*****++
-- Iss 01:01 12-Sep-2001 DRT Created from requirement documents
-- Iss 01:02 1-Oct-2001 DRT Format changes to fit an A4 page
-- Iss 01:03 9-Oct-2001 DRT softReset & batREPstatus added,
--          most asyncCON objects now read-write.
-- Iss 01:04 9-Oct-2001 DRT batCONcheck, phoneNumber and
--          remoteOnline added
-- Iss 01:05 17-Oct-2001 DRT Conversions to SNMPv2, using a strict
--          compiler (SMICng on PC) for syntax checking
-- Iss 01:06 26-Nov-2001 DRT gwConActive change to Unsigned32
-- Iss 01:07 11-Apr-2002 DRT dxCONbit renamed to dxCONbit.
--          Changed the format and extended the function
--          of the various filename fields in the configuration
--          section. File names are now 12 characters using the
--          DOS 8.3 format. The extension will identify the file
--          type, needed for validation checks.
--          Proposed extensions are:-
--          .INI - configuration file
--          .PLN - plan file

```

```

--          .TBL - Time-table file
--          .SCH - Weekly Schedule file
--          other extensions are reserved for future use.
-- lss 01:08 16-Apr-2002 DRT      Added 'deleteFile' OID, plus found a way to
--                               use Unsigned32 on the AXP compiler, use Gauge32!
--                               Description of hcCONbit changed from demand to inhibit
--                               added missing objects fICONbit, dREPbits, tsREPbit,
--                               doREPbit, ehREPbit, evREPbit, vgREPbit, lmuREPbits &
--                               fiREPbit. lrtCONDemand & lrtREPphasereply changed to
--                               bit masks (Gauge32)
-- lss 01:09 5-Jun-2002 DRT      Restored 'proprietary' control and reply objects
-- lss 01:10 14-Jun-2002 DRT      Added timeserverIP, opMode, ttName and
--                               flowThreshold
-- lss 01:11 16-Aug-2002 DRT      sSOS objects moved to Configuration section
-- lss 01:12 05-Sep-2002 DRT      Definition and description of the sSOS objects
--                               clarified

```

-- SCOOT OBJECTS

UTMC29-MIB                    DEFINITIONS ::= BEGIN

IMPORTS

```

Integer32,
Gauge32,
OBJECT-TYPE,
MODULE-IDENTITY
    FROM SNMPv2-SMI

```

```

OBJECT-GROUP
    FROM SNMPv2-CONF

```

```

utmc29
    FROM NEMA-SMI;

```

```

--          This MIB contains three primary sections, these being
--          1/ Configuration control
--          2/ Control functions (SET messages) that have an asynchronous
--             response beyond a predictable timeout
--          3/ Asynchronous events in the OTU

```

```

--          for items in groups 1 & 2, the UTC system is the Manager
--          for items in group 3, the I-OUT is the Manager
--          or a sub-set of group 3 can be polled by the UTC system

```

```

--          This MIB will reflect this structure by defining the primary
--          branches

```

```

--          configuration(1)
--          async-CON(2)
--          async-EVE(3)

```

```

utmc29Module                    MODULE-IDENTITY
    LAST-UPDATED "0209050950Z"
    ORGANIZATION "Peek Ltd.

```

Kings Worthy,  
Winchester  
Hampshire  
SO23 7QA"

CONTACT-INFO "D.R.Tate"  
DESCRIPTION "1:10  
Developed for the UTMC29 project as the interface with an I-OUT"

REVISION "0209050950Z"  
DESCRIPTION  
"sSOsec & sSOStatus objects Definition and Description clarified"

REVISION "0208161105Z"  
DESCRIPTION  
"sSOsec & sSOStatus objects moved to Configuration, where they should be"

REVISION "0206141045Z"  
DESCRIPTION  
"Configuration objects Added timeserverIP, opMode, ttName plus Event object flowThreshold added"

REVISION "0206051652Z"  
DESCRIPTION  
"Support for the sSOsec & sSOStatus objects added in control and event branches"

REVISION "0204161634Z"  
DESCRIPTION  
"Added 'deleteFile' OID, plus found a way to use Unsigned32 on the AXP compiler, use Gauge32!  
Description of hcCONbit changed from demand to inhibit  
Added missing objects flCONbit, dREPbits, tsREPbit, doREPbit, ehREPbit, evREPbit, vgREPbit, lmuREPbits & flREPbit.  
lrtCONDemand & lrtREPphasereply changed to bit masks (Gauge32)"

REVISION "0204111053Z"  
DESCRIPTION  
"dxCONbit renamed to dxCONbit.  
Changed the format and extended the function of the various filename fields in the configuration section.  
File names are now 12 characters using the DOS 8.3 format.  
The extension will identify the file type, needed for validation checks.  
Proposed extensions are:-  
  .CFG - configuration file  
  .PLN - plan file  
  .TBL - Time-table file  
  .SCH - Weekly Schedule  
other extensions are reserved for future use."

REVISION "0110181218Z"  
DESCRIPTION "Iss 1:06  
Initial formal release for the UTMC29 project"  
::= { utmc29 1 }

```

configuration          OBJECT-GROUP
    OBJECTS            {
-- configuration OBJECTS
    fileName,
    fName,
    fCheck,
    loadFile,
    phoneNumber,
    instationIP,
    softReset,
    batteryCheck,
    batteryStatus,
    remoteOnline,
    deleteFile,
    timeserverIP,
    opMode,
    ttName,
    sSOsec,
    sSOStatus,
-- control OBJECTS
    conID,
    fCONbits,
    dCONbits,
    dxCONbit,
    vhCONbit,
    pdCONbit,
    sfCONbit,
    vsCONbits,
    loCONbit,
    soCONbit,
    llCONbit,
    lrtCONinhibit,
    lrtCONdemand,
    gwCONonline,
    gwCONreceived,
    gwCONactive,
    beCONpermit,
    tsCONoverride,
    goCONbit,
    csCONbit,
    hcCONbit,
    flCONbit,
-- event OBJECTS
    eveID,
    gREPbits,
    cfREPbit,
    dfREPbit,
    wiREPbit,
    rrREPbit,
    loREPbit,
    sfREPbit,
    lf1REPbit,

```

```

mmREPbit,
lf2REPbit,
vsREPbits,
sdREPbytes,
vcREPbit,
odREPbit,
qdREPbit,
gwREPCancel,
gwREPdetector,
gwREPrequest,
udsfREPbit,
beREPrunning,
lrtREPmode,
lrtREPdetectorsuspect,
lrtREPlampfail,
lrtREPdetectorfail,
lrtREPwatchdogfail,
lrtREPphasereply,
dREPbits,
tsREPbit,
doREPbit,
ehREPbit,
evREPbit,
vgREPbit,
lmuREPbits,
flREPbit,
flowThreshold
}

```

STATUS current

DESCRIPTION

"The list of objects that configure the system"

```
 ::= { utmc29Module 1 }
```

fileName OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(12))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Allows the Client to specify the current configuration 'file' name.

This 'file' will be used in the future as the default until it is

replaced. This file can, and should, be checked before it is used"

```
 ::= { configuration 1 }
```

checkFileTable OBJECT-TYPE

SYNTAX SEQUENCE OF CheckFileEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Requests a consistency check be performed on the identified file.

This should be done as a matter of course once a new file has been

loaded, to double check that the file has been delivered correctly.

Apart from performing the checks, no other actions are required"

```

 ::=      { configuration 2 }

checkFileEntry      OBJECT-TYPE
    SYNTAX      CheckFileEntry
    MAX-ACCESS    not-accessible
    STATUS      current
    DESCRIPTION  ""
    INDEX       { fName }
 ::=      { checkFileTable 1 }

CheckFileEntry      ::= SEQUENCE {
    fName      OCTET STRING,
    fCheck     Integer32
}

fName              OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(12))
    MAX-ACCESS    read-write
    STATUS      current
    DESCRIPTION  "The name of the file to be checked"
 ::=      { checkFileEntry 1 }

fCheck             OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS    read-write
    STATUS      current
    DESCRIPTION  "The check value that is used for comparison purposes. The local
check must match this value for the file to be considered 'good'.
A 'good' response implies the file checks out and could be used.
A 'bad' response implies the file does not exist, or is unusable"
 ::=      { checkFileEntry 2 }

loadFile           OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(12))
    MAX-ACCESS    read-write
    STATUS      current
    DESCRIPTION  "Requests the remote device abandons its current configuration and
load the configuration file specified. In a perfect world, this
file would have been checked for validity before this request is made.
Loading a configuration file does not make it the default,
only the current.
This allows a new, temporary, configuration to be tested whilst keeping
a good known configuration in reserve in case anything goes wrong!"
 ::=      { configuration 3 }

phoneNumber       OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(20))
    MAX-ACCESS    read-write
    STATUS      current
    DESCRIPTION

```

"Supplies the OTU with the phone number of the Host.  
For security, the value cannot be read."

::= { configuration 4 }

instationIP                    OBJECT-TYPE  
    SYNTAX                    OCTET STRING (SIZE(8))  
    MAX-ACCESS                read-write  
    STATUS                    current  
    DESCRIPTION

"If used, this gives the IP address of the host system.  
Most circumstances will not need this, only envisiaged  
for a dial-up system.

For security, the value cannot be read."

::= { configuration 5 }

softReset                    OBJECT-TYPE  
    SYNTAX                    Integer32  
    MAX-ACCESS                read-write  
    STATUS                    current  
    DESCRIPTION

"Request the OTU to do a software reset. The value is used and  
checked for correctness before the reset is applied.

For security, the reset value cannot be read."

::= { configuration 6 }

batteryCheck                OBJECT-TYPE  
    SYNTAX                    Integer32 (0..1)  
    MAX-ACCESS                read-write  
    STATUS                    current  
    DESCRIPTION

"Request the OTU to do a battery check. As this may take some time  
to complete, get the result via batREPstatus.

If this is read, it gives the state of a battery check.

'0' if not-in-progress, or '1' if in-progress"

::= { configuration 7 }

batteryStatus                OBJECT-TYPE  
    SYNTAX                    INTEGER {  
                  good                    (1),  
                  fail                    (2),  
                  powerfail (3)  
                  }  
    MAX-ACCESS                read-write  
    STATUS                    current  
    DESCRIPTION

"Gives the result of a battery test. The test may be scheduled  
or requested"

::= { configuration 8 }

remoteOnline                OBJECT-TYPE  
    SYNTAX                    Integer32 (0..1)  
    MAX-ACCESS                read-write

```

        STATUS          current
        DESCRIPTION
        "Gives the Outstation the opportunity to notify the Host that
        it is alive and well, and would like a control message"
        ::=             { configuration 9 }

deleteFile          OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE(12))
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
        "Requests the server to delete a previously down-loaded file"
        ::=             { configuration 10 }

timeserverIP       OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE(8))
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
        "If used, this gives the IP address of the timeserver.
        For security, the value cannot be read."
        ::=             { configuration 11 }

opMode             OBJECT-TYPE
        SYNTAX          INTEGER {
                standalone (1),
                monitor     (2),
                utcccontrol (3)
                }
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
        "Tells the Outstation what mode to operate in"
        ::=             { configuration 12 }

ttName             OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE(12))
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
        "Allows the Client to specify /request the current Time-Table 'file' name."
        ::=             { configuration 13 }

sSOsec            OBJECT-TYPE
        SYNTAX          Gauge32
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
        "Used for second-by-second control.
        The Gauge32 value maps directly onto the 32 output bits"
        ::=             { configuration 14 }

```

sSOStatus OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(8))  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"Provide 32 controller data bits, mapped directly from the 32 inputs, and Scoot detector data as the previous four quarter second scans. The controller data is in the first four Octets, with the lower numbered bits in the first Octet. An open-circuit input is transmitted as a 1 Scoot data is in the last four Octets with the lower numbered SCOOT detectors are in the first of the four Octets with the lower detector in the lower nibble. The most recent detector sample is in the l.s.b. of the nibble. Presence is indicated with a 1"

::= { configuration 15 }

conObjectsTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF ConObjectsEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION

"These are all the possible control functions that the UTC can use"

::= { utmc29Module 2 }

conObjectsEntry OBJECT-TYPE  
 SYNTAX ConObjectsEntry  
 MAX-ACCESS not-accessible  
 STATUS current  
 DESCRIPTION ""  
 INDEX { conID }  
 ::= { conObjectsTable 1 }

ConObjectsEntry ::= SEQUENCE {  
 conID OCTET STRING,  
 fCONbits Gauge32,  
 dCONbits Gauge32,  
 dxCONbit Integer32,  
 vhCONbit Integer32,  
 pdCONbit Integer32,  
 sfCONbit Integer32,  
 vsCONbits Integer32,  
 loCONbit Integer32,  
 soCONbit Integer32,  
 lICONbit Integer32,  
 lrtCONinhibit Integer32,  
 lrtCONdemand Gauge32,  
 gwCONonline Integer32,  
 gwCONreceived Integer32,  
 gwCONactive Gauge32,  
 beCONpermit Integer32,  
 tsCONoverride Integer32,  
 goCONbit Integer32,

```

csCONbit      Integer32,
hcCONbit      Integer32,
fCONbit Integer32
}

```

```

conID          OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(6))
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

"Identifies which, of possibly several, equipment at this site the object should be applied to.

This could be an SCN, IP address, or a number"

```
 ::= { conObjectsEntry 1 }
```

```

fCONbits       OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

"A bit map of stage force bits"

```
 ::= { conObjectsEntry 2 }
```

```

dCONbits       OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

"A bit map of stage demand bits"

```
 ::= { conObjectsEntry 3 }
```

```

dxCONbit       OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

"Common demand for all stages"

```
 ::= { conObjectsEntry 4 }
```

```

vhCONbit       OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION

```

"Pedestrian-crossing vehicle hold"

```
 ::= { conObjectsEntry 5 }
```

```

pdCONbit       OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""

```

```

 ::=      { conObjectsEntry 6 }

sfCONbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
 ::=      { conObjectsEntry 7 }

vsCONbits     OBJECT-TYPE
    SYNTAX      Integer32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
 ::=      { conObjectsEntry 8 }

loCONbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
    "Lamps On/Off control bit"
 ::=      { conObjectsEntry 9 }

soCONbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
    "Solar OverRide control"
 ::=      { conObjectsEntry 10 }

lICONbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
 ::=      { conObjectsEntry 11 }

IrtCONinhibit OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
 ::=      { conObjectsEntry 12 }

IrtCONdemand  OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION ""
 ::=      { conObjectsEntry 13 }

```

```

gwCONonline          OBJECT-TYPE
    SYNTAX            Integer32 (0..1)
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "A signal to the Green Wave Panel that the system is on-line"
    ::=               { conObjectsEntry 14 }

gwCONreceived        OBJECT-TYPE
    SYNTAX            Integer32 (0..1)
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "Confirmation that the last Green Wave action has been accepted"
    ::=               { conObjectsEntry 15 }

gwCONactive          OBJECT-TYPE
-- 01:06 changed SYNTAX from OCTET STRING to Unsigned32, then Gauge32
    SYNTAX            Gauge32
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "A bit-map (8 bits only) of all the currently active Green Waves"
    ::=               { conObjectsEntry 16 }

beCONpermit          OBJECT-TYPE
    SYNTAX            Integer32 (0..1)
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "Bus Extension Permit"
    ::=               { conObjectsEntry 17 }

tsCONoverride        OBJECT-TYPE
    SYNTAX            Integer32 (0..1)
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "Time-Switch Override"
    ::=               { conObjectsEntry 18 }

goCONbit             OBJECT-TYPE
    SYNTAX            Integer32 (0..1)
    MAX-ACCESS        read-write
    STATUS            current
    DESCRIPTION
    "Gap Out"
    ::=               { conObjectsEntry 19 }

csCONbit             OBJECT-TYPE
    SYNTAX            Integer32 (0..1)

```

```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
"Clock Synch Bit"
 ::=           { conObjectsEntry 20 }

hcCONbit        OBJECT-TYPE
SYNTAX          Integer32 (0..1)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
"Hurry Call inhibit bit"
 ::=           { conObjectsEntry 21 }

fICONbit        OBJECT-TYPE
SYNTAX          Integer32 (0..1)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
"Flashing Request Bit bit"
 ::=           { conObjectsEntry 22 }

eveObjectTable  OBJECT-TYPE
SYNTAX          SEQUENCE OF EveObjectEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
"These are all the possible reply events that the UTC can expect.
For these objects,
EITHER
the I-OUT is the manager and spontaneously sends the objects
OR
the UTC system polls a sub-set of the objects as the manager
BUT NOT BOTH!"
 ::=           { utmc29Module 3 }

eveObjectEntry  OBJECT-TYPE
SYNTAX          EveObjectEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION     ""
INDEX           { eveID }
 ::=           { eveObjectTable 1 }

EveObjectEntry ::= SEQUENCE {
eveID           OCTET STRING,
gREPbitsGauge32 Integer32,
cfREPbit Integer32,
dfREPbit Integer32,
wiREPbit Integer32,
rrREPbit Integer32,
loREPbit Integer32,

```

```

sfREPbit Integer32,
lf1REPbitInteger32,
mmREPbit Integer32,
lf2REPbitInteger32,
vsREPbits Integer32,
sdREPbytes OCTET STRING,
vcREPbit Integer32,
odREPbitInteger32,
qdREPbitInteger32,
gwREPCancel Integer32,
gwREPdetector Integer32,
gwREPrequest Gauge32,
udsfREPbit Integer32,
beREPrunning Integer32,
lrtREPmode Integer32,
lrtREPdetectorsuspect
Integer32,
lrtREPlampfail Integer32,
lrtREPdetectorfail
Integer32,
lrtREPwatchdogfail
Integer32,
lrtREPphasereply
Gauge32,
dREPbitsGauge32,
tsREPbit Integer32,
doREPbitInteger32,
ehREPbitInteger32,
evREPbitInteger32,
vgREPbitInteger32,
lmuREPbits Gauge32,
flREPbit Integer32,
flowThreshold INTEGER
}

```

```

eveID OBJECT-TYPE
SYNTAX OCTET STRING (SIZE(6))
MAX-ACCESS read-write
STATUS current
DESCRIPTION

```

"Identifies which, of possibly several, equipment at this site the object should be applied to.

This could be an SCN, IP address, or a number"

```
 ::= { eveObjectEntry 1 }
```

```

gREPbits OBJECT-TYPE
SYNTAX Gauge32
MAX-ACCESS read-write
STATUS current
DESCRIPTION

```

"A bit map of green confirm bits"

```
 ::= { eveObjectEntry 2 }
```

cfREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Controller Fault bit"  
 ::= { eveObjectEntry 3 }

dfREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Detector Fault bit"  
 ::= { eveObjectEntry 4 }

wiREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Wait Indicator"  
 ::= { eveObjectEntry 5 }

rrREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Remote Reconnect"  
 ::= { eveObjectEntry 6 }

loREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Lamps Off"  
 ::= { eveObjectEntry 7 }

sfREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Special Facility"  
 ::= { eveObjectEntry 8 }

If1REPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write

STATUS current  
 DESCRIPTION  
 "Lamp Fail (1)"  
 ::= { eveObjectEntry 9 }

mmREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Manual Mode"  
 ::= { eveObjectEntry 10 }

lf2REPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Lamp Fail (2)"  
 ::= { eveObjectEntry 11 }

vsREPbits OBJECT-TYPE  
 SYNTAX Integer32  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "Variable Sign"  
 ::= { eveObjectEntry 12 }

sdREPbytes OBJECT-TYPE  
 SYNTAX OCTET STRING (SIZE(4))  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"These contain the SCOOT detector data. This version presents the data in the same format as in the MCE0361 comm. packet. That is, the lower numbered SCOOT detectors are in the first Octet with the lower detector in the lower nibble. The most recent detector sample is in the l.s.b. of the nibble. It may be more useful to format the data as one detector per byte of the string?"

::= { eveObjectEntry 13 }

vcREPbit OBJECT-TYPE  
 SYNTAX Integer32 (0..1)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION

"Vehicle Count Detector"  
 ::= { eveObjectEntry 14 }

odREPbit OBJECT-TYPE

```

        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Occupancy Detector bit"
        ::=             { eveObjectEntry 15 }

qdREPbit              OBJECT-TYPE
        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Queue Detector bit"
        ::=             { eveObjectEntry 16 }

gwREPCancel           OBJECT-TYPE
        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Sent by the OTU when the button is pushed"
        ::=             { eveObjectEntry 17 }

gwREPdetector         OBJECT-TYPE
        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Sent by the OTU when the detector is activated"
        ::=             { eveObjectEntry 18 }

gwREPrequest          OBJECT-TYPE
        SYNTAX          Gauge32
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Sent by the OTU, bit mask of buttons pushed"
        ::=             { eveObjectEntry 19 }

udsfREPbit            OBJECT-TYPE
        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION
"Sent by the OTU when the UDSF bit changes state"
        ::=             { eveObjectEntry 20 }

beREPRunning          OBJECT-TYPE
        SYNTAX          Integer32 (0..1)
        MAX-ACCESS      read-write
        STATUS          current
        DESCRIPTION

```

```

"Bus Extension is running"
    ::=      { eveObjectEntry 21 }

IrtREPmode          OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    "Irt mode, ON/OFF"
    ::=      { eveObjectEntry 22 }

IrtREPdetectorsuspect OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    ""
    ::=      { eveObjectEntry 23 }

IrtREPlampfail     OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    ""
    ::=      { eveObjectEntry 24 }

IrtREPdetectorfail OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    ""
    ::=      { eveObjectEntry 25 }

IrtREPwatchdogfail OBJECT-TYPE
    SYNTAX          Integer32 (0..1)
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    ""
    ::=      { eveObjectEntry 26 }

IrtREPphasereply  OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
    "A Bit Mask of LRT phase replies"
    ::=      { eveObjectEntry 27 }

dREPbits          OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION

```

```

"Detector reply bit mask"
    ::=      { eveObjectEntry 28 }

tsREPbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Time Synch Confirm reply bit"
    ::=      { eveObjectEntry 29 }

doREPbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Door Open reply bit"
    ::=      { eveObjectEntry 30 }

ehREPbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Explosion Hazard reply bit"
    ::=      { eveObjectEntry 31 }

evREPbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Emergency Vehicle reply bit"
    ::=      { eveObjectEntry 32 }

vgREPbit      OBJECT-TYPE
    SYNTAX      Integer32 (0..1)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"Pelican Vehicle Green reply bit"
    ::=      { eveObjectEntry 33 }

ImuREPbits    OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
"The reply from a Lamp Monitor"
    ::=      { eveObjectEntry 34 }

flREPbit      OBJECT-TYPE

```

```
SYNTAX          Integer32 (0..1)
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
"Flashing Confirm reply bit"
 ::=           { eveObjectEntry 35 }

flowThreshold   OBJECT-TYPE
SYNTAX          INTEGER {
                upper    (1),
                lower    (2)
                }
MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
"Warning that the detector flow threshold value (upper or lower) has been passed"
 ::=           { eveObjectEntry 36 }

END
```

**E.7 UM/029, Car Park Monitor MIB**

UTMC-CarParks DEFINITIONS ::= BEGIN

```

--      Y1-04011.txt
--      Revision: 2.01
--      Product No:      Car Park Monitor
--      Date:            22/2/2005
--      Written: Robin Jefferson

--      Revision History
--      V1.00  30/5/2002  First Issue                                RLJ
--      V1.01  31/5/2002  Addition of a trap to indicate a fault          RLJ
--      V1.02  31/5/2002  Change of AQM reference                          RLJ
--      V1.03  25/6/2002  Change carParkOccupancy to Read-Write to allow reset RLJ
--                                     Change of Queue length to optional
--                                     Add Auto-reset and reset value objects
--                                     Enumurate carParkOccupancyTrend
--      V1.04  2/8/2002   Correction to traps, and addition of quote marks          RLJ
--      V1.05  5/2/2003   Addition of port and IP address                          RLJ
--                                     Addition of Historical data
--                                     Addition of uni-directional loop configuration
--      V1.06  16/6/2003   Add 'Unassigned' to loops                                RLJ
--      V1.07  16/10/2003  Add start loop assignments to counter table           RLJ
--      V1.08  19/5/2004   Add object 'carParkMIBVer' - MIB version             RLJ
--                                     Change 'carParkDataFillRate' and 'carParkDataExitRate'
--                                     from vehicles/min to absolute vehicle count
--                                     Add object 'carParkDataCounters' Returns data for
--                                     individual loops/counters
--      V2.01  18/2/05    Modifications following harmonisation
--      V2.01  22/2/05    Mods following review

--      City of York Council
--      9 St Leonards Place
--      York
--      YO1 7ET
--      Tel +44 1904 551372
--      Fax +44 1904 551412

--      Maintained by
--      Integrated Design Techniques Ltd
--      Endurance House
--      Seventh Avenue
--      Team Valley
--      Tyne & Wear
--      NE11 0EF
--      Tel +44 191 491 0800
--      Fax +44 191 491 0799
--      email:  robin@idtuk.com

```

```
--      This module provides definitions and registration points for
--      City of York Council's UTMC compliant Car Park Monitors

--      City of York Council reserve the right to make changes in this specification
--      and other information contained in this document without
--      prior notice. In no event shall City of York Council be liable for any
--      incidental, indirect, special or consequential damages arising out of, or
--      related to the use of this document or the information contained in it.
--      City of York Council grant vendors and end-users a non-exclusive
--      licence to use this specification in the connection with management
--      of UTMC compliant outstations.

--      Copyright City of York Council 2002
```

IMPORTS

```
    TRAP-TYPE
        FROM RFC-1215
    OBJECT-TYPE
        FROM RFC-1212
        utmc, utmcCarParksType1, UTMCTime
            FROM UTMC-Header-MIB;
```

```
    SMI IpAddress
```

```
--      Textual conventions
--Mod V2.0 - Remove and insert in header MIB
--DisplayString ::= OCTET STRING
--      This data type is defined to support textual information using
--      the ASCII character set. By convention, objects declared with this
--      syntax, unless otherwise specified are declared as having:
--
--      SIZE (0..255)
```

```
--      the path to the root
```

```
    carParkSystem            OBJECT IDENTIFIER ::= { utmcCarParksType1 1 }
    carParkZone             OBJECT IDENTIFIER ::= { utmcCarParksType1 2 }
    carParkCounter         OBJECT IDENTIFIER ::= { utmcCarParksType1 3 }
    carParkFault            OBJECT IDENTIFIER ::= { utmcCarParksType1 4 }
    carParkData             OBJECT IDENTIFIER ::= { utmcCarParksType1 5 }
```

```
-----
-- Traps
-----
```

```
carParkStateTrap TRAP-TYPE
    ENTERPRISE utmc
    VARIABLES      { carParkState }
    DESCRIPTION
        "This trap returns the objects carparkState when a threshold is exceeded"

    ::= 0
```

```

faultAlarm          TRAP-TYPE
                    ENTERPRISE utmc
                    VARIABLES { carParkFaultNo, carParkFaultID, carParkFaultType, carParkFaultDate,
carParkFaultRTC, carParkFaultDesc }
                    DESCRIPTION
                        "This trap indicates that a fault has occurred.
                        The last recorded fault is returned."
                    ::= 1

```

-----  
-- General and Identification objects  
-----

```

carParkSoftwareVer  OBJECT-TYPE
                    SYNTAX DisplayString (SIZE(50))
                    ACCESS read-only
                    STATUS mandatory
                    DESCRIPTION
                        "Returns the Car Park controller Software Version as Vmajor.minor"
                    ::= { carParkSystem 1 }

```

--Mod V2.0 - Add 'logoff' to this object

```

carParkPassword     OBJECT-TYPE
                    SYNTAX DisplayString (SIZE(50))
                    ACCESS read-write
                    STATUS mandatory
                    DESCRIPTION
                        "The password object must be accessed and a valid password entered before any of
the following objects will be available to the SNMP interface. A value of 'logoff' is used to log the user off.
A value of null indicates that no password is been used."
                    ::= { carParkSystem 2 }

```

--Mod V2.0 - Deprecate this object - replaced by 'logoff' in carParkPassword.

```

carParkLogOff       OBJECT-TYPE
                    SYNTAX INTEGER
                    ACCESS read-write
                    STATUS deprecated
                    DESCRIPTION
                        "Logs the user off and requires the password to be re-entered to access all further
objects."
                    ::= { carParkSystem 3 }

```

--Mod V2.0 - Deprecate this object - replaced by carParkRTC

```

carParkTime         OBJECT-TYPE
                    SYNTAX INTEGER
                    ACCESS read-write
                    STATUS deprecated
                    DESCRIPTION
                        "Sets or returns the current time (in seconds since 1st January 1970 00:00:00)."
                    ::= { carParkSystem 4 }

```

carParkIPAddress OBJECT-TYPE

SYNTAX IpAddress  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

" This object hold the IP Address to which traps are returned.  
If the object is invalid or 0.0.0.0 then traps are returned to the IP Address of  
the manager which last made a Set or Get request"  
 ::= { carParkSystem 5 }

carParkPort OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

" This object hold the Port number to which traps are returned.  
If the object is invalid or 0 then traps are returned to the local Port of  
the manager which last made a Set or Get request"  
 ::= { carParkSystem 6 }

carParkMIBVer OBJECT-TYPE

SYNTAX INTEGER  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION

"Returns the Car Park MIB Version. Version 1.08 is sent as 108"  
 ::= { carParkSystem 7 }

--Mod V2.0 - New object to set the password

carParkSetPassword OBJECT-TYPE

SYNTAX DisplayString (SIZE(50))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"This object sets the password to be used to log on. Null indicates no password"  
 ::= { carParkSystem 8 }

--Mod V2.0 - New object to set or read the time

carParkRTC OBJECT-TYPE

SYNTAX UTMCTime  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"This object sets or returns the current time"  
 ::= { carParkSystem 9 }

-----  
-- Zone Configuration and data  
-----

--Mod V2.01 - Change Octet string to displaystring

carParkZoneID OBJECT-TYPE

SYNTAX DisplayString(SIZE(5..32))

ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
     "Zone identifier"  
 ::= { carParkZone 1 }

carParkCapacity OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
     "Capacity of car-park/zone"  
 ::= { carParkZone 2 }

--Mod V2.01 - Make these objects override values

carParkOpeningTime OBJECT-TYPE  
 SYNTAX INTEGER (-1..1440)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
     "The override opening time of the car park/zone time (in minutes since midnight).  
     This object is set to 0 if the car park is 24 hours. If -1 is set the opening and closing  
 times revert to the carParkOpeningTable"  
 ::= { carParkZone 3 }

--Mod V2.01 - Make these override values

carParkClosingTime OBJECT-TYPE  
 SYNTAX INTEGER (-1..1440)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
     "The override closing time of the car park/zone time (in minutes since midnight).  
     This object is set to 0 if the car park is 24 hours. If -1 is set the opening and closing  
 times revert to the carParkOpeningTable"  
 ::= { carParkZone 4 }

carParkAFincreasing OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
     "The threshold for almost full (Occupancy )."  
 ::= { carParkZone 5 }

carParkAFdecreasing OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
     "The threshold below which the car park/zone has spaces (in occupancy)."  
 ::= { carParkZone 6 }

```
carParkFincreasing      OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The threshold at which the car park/zone is full (in occupancy)."
```

::= { carParkZone 7 }
-----------------------

```
carParkFdecreasing    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The value below which the car park/zone becomes almost full."
```

::= { carParkZone 8 }
-----------------------

```
carParkEntranceFull   OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The value at which the car park is full at it's entrance."
```

::= { carParkZone 9 }
-----------------------

```
carParkTrapTrigger    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object sets which state to trigger a trap. A trap will trigger only once on each
entry to trap configured
    state
        b0 (1) - Almost Full Increasing
        b1 (2) - Almost Full Decreasing
        b2 (4) - Full Increasing
        b3 (8) - Full Decreasing
        b4 (16)- Full"
```

::= { carParkZone 10 }
------------------------

--Mod V2.01 - Add spaces and closed as states

```
carParkState          OBJECT-TYPE
    SYNTAX INTEGER {
        almostFullIncreasing(1),
        almostFullDecreasing(2),
        fullIncreasing(3),
        fullDecreasing(4),
        full(5),
        spaces(6),
        closed(7)
    }
    ACCESS read-only
    STATUS mandatory
```

DESCRIPTION

"The current state of occupancy of the car park/zone. This value is updated every minute."

::= { carParkZone 11 }

carParkOccupancy OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The current occupancy (number of cars) of the car park/zone. This value is updated every minute"

::= { carParkZone 12 }

carParkOccPercent OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current occupancy (percentage full) of the car park/zone. This value is updated every minute."

::= { carParkZone 13 }

carParkFillRate OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current fill rate (no of vehicles) of the car park/zone. This value is updated every minute"

::= { carParkZone 14 }

carParkExitRate OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current exit rate (no of vehicles) of the car park/zone. This value is updated every minute."

::= { carParkZone 15 }

carParkOccTrend OBJECT-TYPE

SYNTAX INTEGER {

down(1),

stay(2),

up(3)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The current occupancy trend of the car park/zone (down, stay, up) based upon the last 15 minutes of occupancy values.

This value is updated every minute and is a rolling average"

::= { carParkZone 16 }

carParkQueue OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION

"An estimate of car park/zone queuing time in minutes. This value is updated every minute"

::= { carParkZone 17 }

--Mod V2.01 - Change Autoreset to set a reset time

carParkAutoReset OBJECT-TYPE  
 SYNTAX INTEGER (-1..1440)  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

"Disables or sets the time for automatic reset of the occupancy. A Value of -1 turns off the autoreset feature."

::= { carParkZone 18 }

carParkAutoResetValue OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION

"Occupancy Value for auto-reset"

::= { carParkZone 19 }

--Mod V2.01 - Deprecate this object and use the carParkCounterTable

carParkDualCount OBJECT-TYPE  
 SYNTAX DisplayString (SIZE(5))  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION

"Used to read and write the configuration settings for dual loop counting:-

Format Char1Char2Char3Char4Char5

Char 1            ' 1' = all single loop counting  
                   ' 2' = 1 dual loop count 6 single loop  
                   ' 3' = 2 dual loop counts 4 single loop  
                   ' 4' = 3 dual loop counts 2 single loop  
                   ' 5' = 4 dual loop counts 0 single loop

Chars 2 - 5        ' 1' = bi-directional count     (Char 2 applies to first dual counter)  
                   ' 0' = uni-directional count"

::= { carParkZone 20 }

carParkOpeningTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF CarParkOpeningEntry  
 ACCESS not-accessible  
 STATUS mandatory

DESCRIPTION

"Table contains Opening and closing times for the car park"

::= {carParkZone 21}

--Mod V2.01 - Add atable of opening and closing times

carParkOpeningEntry OBJECT-TYPE

SYNTAX CarParkOpeningEntry

ACCESS not-accessible

STATUS mandatory

DESCRIPTION

"This object is used for configuration of Car Park opening times."

INDEX {carParkOpeningDay}

::= {carParkOpeningTable 1}

CarParkOpeningEntry ::= SEQUENCE {

carParkOpeningDay INTEGER,  
 carParkOpeningOpen INTEGER,  
 carParkOpeningClose INTEGER,  
 }

carParkOpeningDay OBJECT-TYPE

SYNTAX INTEGER {  
 monday(1),  
 tuesday(2),  
 wednesday(3),  
 thursday(4),  
 friday(5),  
 saturday(6),  
 sunday(7)  
 }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Day of the week"

::={carParkOpeningEntry 1}

carParkOpeningOpen OBJECT-TYPE

SYNTAX INTEGER (0..1440)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Opening time in minutes past midnight"

::={carParkOpeningEntry 2}

carParkOpeningClose OBJECT-TYPE

SYNTAX INTEGER (0..1440)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Closing time in minutes past midnight"

::={carParkOpeningEntry 3}

-----  
 -- Counter Configuration  
 -----

carParkCounterTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF CarParkCounterEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "Table contains unit configuration with respect to monitored counters."  
 ::= {carParkCounter 1}

carParkCounterEntry OBJECT-TYPE  
 SYNTAX CarParkCounterEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "This object is used for configuration values relating to a counter."  
 INDEX {carParkCounterNum}  
 ::= {carParkCounterTable 1}

CarParkCounterEntry ::= SEQUENCE {  
     carParkCounterNum INTEGER,  
     carParkCounterEntryExit INTEGER,  
     carParkNumCounters INTEGER,  
     carParkCounterStartLoop INTEGER,  
     carParkCounterLoopType INTEGER  
 }

carParkCounterNum OBJECT-TYPE  
 SYNTAX INTEGER (1..64)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "counter number (1->N)"  
 ::= {carParkCounterEntry 1}

carParkCounterEntryExit OBJECT-TYPE  
 SYNTAX INTEGER {  
     eNTRY(1),  
     eXIT(2),  
     uNASSIGNED(3)  
 }  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "States if the counter is an entry or exit counter."  
 ::= { carParkCounterEntry 2 }

--Mod V2.01 - Deprecate

```

carParkNumCounters OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "The number of counters on a unit."
    ::= { carParkCounterEntry 3 }

carParkCounterStartLoop OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "The detector number of the first detector connected to this counter."
    ::= { carParkCounterEntry 4 }

carParkCounterLoopType OBJECT-TYPE
    SYNTAX INTEGER {
        single(1),
        unidirectional(2),
        bidirectional(3)
    }
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The type of detector connected on this input."
    ::= { carParkCounterEntry 5 }

```

-----  
-- Faults  
-----

--Mod V2.01 - Add a fault string, detector Fail (DF) and real time

```

carParkFaultTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CarParkFaultEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "Provides a table of instances of faults."
    ::= { carParkFault 1 }

```

```

carParkFaultEntry OBJECT-TYPE
    SYNTAX CarParkFaultEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of faults"
    INDEX { carParkFaultNo }
    ::= { carParkFaultTable 1 }

```

```

CarParkFaultEntry ::=
    SEQUENCE {

```

```

carParkFaultNo
    INTEGER,
carParkFaultID
    INTEGER,
carParkFaultType
    INTEGER,
carParkFaultDate
    INTEGER,
carParkFaultRTC
    UTMCTime,
carParkFaultDesc
    OCTET STRING
}

```

```

carParkFaultNo OBJECT-TYPE
    SYNTAX INTEGER (0..256)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Returns the index for a fault entry"
    ::= { carParkFaultEntry 1}

```

```

carParkFaultID OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Returns the identifier for the fault type"
    ::= { carParkFaultEntry 2}

```

```

carParkFaultType OBJECT-TYPE
    SYNTAX INTEGER {
        equipment(1),
        comms(2),
        dF(3)
    }
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "Returns the sub-type for the fault type"
    ::= { carParkFaultEntry 3}

```

--Mod V2.01 - Deprecated and use the carParkRTC object

```

carParkFaultDate OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS deprecated
    DESCRIPTION
        "Returns the time & date for the fault in seconds since 1st Jan 1970."
    ::= { carParkFaultEntry 4}

```

--Mod V2.0 - New object to read the fault time

```

carParkFaultRTC OBJECT-TYPE

```

```

SYNTAX UTMCTime
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "This object returns the fault time"
 ::= { carParkFaultEntry 5}

```

```

carParkFaultDesc OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Returns a user string with extended fault information."
 ::= { carParkFaultEntry 6}

```

-----  
-- Historical Data  
-----

--Mod V2.01 - Deprecate and use carParkDataStartRTC object

```

carParkDataStartTime OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS deprecated
DESCRIPTION
    "Sets the start time in seconds since 1/1/1970 00:00 for download of data"
 ::= { carParkData 1 }

```

```

carParkDataPeriod OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Sets the period in minutes for download of data.
    Only certain periods may be supported in which case the traffic counter
    will default to the last supported value and return a badValue error. As a minimum 5,
    15 & 60 minutes must be supported"
 ::= { carParkData 2 }

```

```

carParkDataOccupancy OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "An average of the occupancy (number of cars) of the car park/zone over the time period
    specified."
 ::= { carParkData 3 }

```

```

carParkDataOccPercent OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION

```

"An average of the occupancy (percentage full) of the car park/zone over the time period specified."

::= { carParkData 4 }

carParkDataFillRate OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The fill rate (number of cars entering the car park) over the time period specified."

::= { carParkData 5 }

carParkDataExitRate OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The exit rate (number of cars leaving the car park) over the time period specified."

::= { carParkData 6 }

carParkDataOccTrend OBJECT-TYPE

SYNTAX INTEGER {

down(1),

stay(2),

up(3)

}

ACCESS read-only

STATUS mandatory

DESCRIPTION

"The average occupancy trend of the car park/zone (down, stay, up) based upon the last 15 minutes of occupancy values over the time period specified."

::= { carParkData 7 }

carParkDataQueue OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-only

STATUS optional

DESCRIPTION

"An estimate of car park/zone queuing time in minutes over the time period specified."

::= { carParkData 8 }

carParkDataCounters OBJECT-TYPE

SYNTAX DisplayString (SIZE(55))

ACCESS read-only

STATUS optional

DESCRIPTION

"Returns the counts over the specified start time and period for the individual loop counters as readable text (First 8).

A positive value is a fill with a negative value an exit.

Bi-directional loops occupy two adjacent fields with a positive and negative count.

Uni-directional loops also occupy two adjacent fields but only one field with have data.

Example: A Uni-direction entrance, followed by two single exits:

+00032,-00000,-00004,-00060,+00000,+00000,+00000,+00000"  
 ::= { carParkData 9 }

--Mod V2.0 - New object to set or read the time

carParkDataStartRTC OBJECT-TYPE  
 SYNTAX UTMCTime  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 "This object sets or returns the fault time"  
 ::= { carParkData 10 }

carParkDataCounterSetUp OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS optional  
 DESCRIPTION  
 "A 32 bit field of counters to return data for over the specified start date and period."  
 ::= { carParkData 11 }

carParkDataIndividualCounts OBJECT-TYPE  
 SYNTAX DisplayString  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
 "Returns the counts over the specified start time and period for the individual loop  
 counters as readable text and as specified by carParkCounterSetUp, with As many  
 values are returned as requested.  
 A positive value is a fill with a negative value an exit.  
 Bi-directional loops occupy two adjacent fields with a positive and negative count.  
 Uni-directional loops also occupy two adjacent fields but only one field with have data.  
 Example: A Uni-direction entrance, followed by two single exits:  
 +00032,-00000,-00004,-00060."  
 ::= { carParkData 12 }

END

**E.8 UM/030, Traffic Counter MIB**

UTMC-TrafficCounter DEFINITIONS ::= BEGIN

```
-- V3-01033.txt
-- Revision: 4.01
-- Product No:      Traffic Counter
-- Date:            22/2/2005
-- Written: Robin Jefferson

-- Revision History
-- Draft V3.00      10/10/2002      Redrawn from Blue Box MIB Rev 2.01 (Y1-02018)
-- V3.01           8/8/2003 Redrawn from Y1-02022 to add dual loops
-- V3.02           16/10/2003      Add Dual loop positions
-- V4.01           22/2/2005      Modifications following MIB harmonisation

-- City of York Council
-- 9 St Leonard's Place
-- York
-- YO1 7ET
-- Tel +44 1904 551372
-- Fax +44 1904 551412

-- Maintained by
-- Integrated Design Techniques Ltd
-- Endurance House
-- Seventh Avenue
-- Team Valley
-- Tyne & Wear
-- NE11 0EF
-- Tel +44 191 491 0800
-- Fax +44 191 491 0799
-- email: robin@idtuk.com

-- This module provides definitions and registration points for
-- City of York Council's UTMC compliant Traffic Counters

-- City of York Council reserve the right to make changes in this specification
-- and other information contained in this document without
-- prior notice. In no event shall City of York Council be liable for any
-- incidental, indirect, special or consequential damages arising out of, or
-- related to the use of this document or the information contained in it.
-- City of York Council grant vendors and end-users a non-exclusive
-- licence to use this specification in the connection with management
-- of UTMC compliant outstations.

-- Copyright City of York Council 2002
```

IMPORTS

OBJECT-TYPE

FROM RFC-1212

TRAP-TYPE

FROM RFC-1215

IpAddress

FROM RFC1155-SMI

utmc, utmcTrafficCounterType1, DisplayString, TruthValue, UTMCTime

FROM UTM-Header-MIB;

-- Textual conventions

--Mod V4.01 - Remove to header mib

--DisplayString ::= OCTET STRING

-- This data type is defined to support textual information using  
 -- the ASCII character set. By convention, objects declared with this  
 -- syntax, unless otherwise specified are declared as having:

-- SIZE (0..255)

-----  
 -- Notes  
 -----

-- Direction/Counters

-- The configuration objects for flow, occupancy and queue and data objects for  
 -- flow, occupancy, speed, classification and queue are valid for up to 8 directions or Counters.  
 -- The direction is set by the trafficCounterSetDirection object prior to access.

-- Speed

-- Speed is that average vehicle speed over the period specified.

-- Queue

-- A queue is detected by monitoring a specific detector. If the detector is occupied  
 -- for a minimum time (known as the call time) then a queue is assumed to have formed.  
 -- The queue detector must be un-occupied for a minimum time (known as the cancel time) before  
 -- the queue is assumed to have cleared. This is to allow for vehicles moving over the  
 -- queue detector and prevents false triggering.

-- Alarms

-- A trap is to be sent when an alarm condition is true and when it has been cleared.

-----  
 -- the path to the root  
 -----

trafficCounterSystem OBJECT IDENTIFIER ::= { utmcTrafficCounterType1 1 }

trafficCounterConfig OBJECT IDENTIFIER ::= { utmcTrafficCounterType1 2 }

trafficCounterData OBJECT IDENTIFIER ::= { utmcTrafficCounterType1 3 }

trafficCounterFault OBJECT IDENTIFIER ::= { utmcTrafficCounterType1 4 }

trafficCounterDual OBJECT IDENTIFIER ::= { utmcTrafficCounterType1 5 }

```

-----
--      Trap definitions
-----
--Mod V4.01 - Change to allow 8 directions over x minutes
trafficCounterAlarm TRAP-TYPE
    ENTERPRISE utmc
    VARIABLES      {trafficCounterFlow,      trafficCounterOccupancy,      trafficCounterSpeed,
trafficCounterAlarmDirection}
    DESCRIPTION
        "This trap returns the x minute flow and occupancy for chosen direction when an
alarm occurs"
        ::= 0

trafficCounterQueueAlarm TRAP-TYPE
    ENTERPRISE utmc
    VARIABLES      {trafficCounterQueue, trafficCounterAlarmDirection}
    DESCRIPTION
        "This trap returns the Queue status"
        ::= 1

--Mod V4.01 - Remove trafficCounterFaultType, change FaultDate to FaultRTC
trafficCounterFaultAlarm TRAP-TYPE
    ENTERPRISE utmc
    VARIABLES      {trafficCounterFaultNo,      trafficCounterFaultID,      trafficCounterFaultRTC,
trafficCounterFaultDirection}
    DESCRIPTION
        "This trap Indicates that a new fault has occurred"
        ::= 2

-----
-- General and Identification objects
-----
--Mod V4.01 - Define format for version
trafficCounterSoftwareVer OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Returns the Traffic Counter Software Version as Vx.y"
        ::= { trafficCounterSystem 1 }

--Mod V4.01 - Deprecate
trafficCounterExitProgram OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "Terminates the Outstation program and casues the traffic counter to go to a
maintenance mode"
        ::= { trafficCounterSystem 2 }

```

```
--Mod V4.01 - Deprecate and use trafficCounterRTC
trafficCounterTime OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "Sets or returns the current time (in seconds since 1st January 1970 00:00:00) on the
Traffic Counter."
    ::= { trafficCounterSystem 3 }
```

```
--Mod V4.01 - Deprecate as no longer thought to be required
trafficCounterLog OBJECT-TYPE
    SYNTAX INTEGER {
        oFF(1),
        oN(2)
    }
    ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "Turns logging of detector data to on or off."
    ::= { trafficCounterSystem 4 }
```

```
--Mod V4.01 - Add RTC
trafficCounterRTC OBJECT-TYPE
    SYNTAX UTMCTime
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Sets or returns the current date/time in the standard format."
    ::= { trafficCounterSystem 5 }
```

-----  
-- Configuration  
-----

```
--Mod V4.01 - Deprecate and use table
trafficCounterLoopConfig OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "This object set or returns the detector configuration. Configuration is set as follows:
```

Bit 0-2 - Lane Configuration Dir 1 | Dir 2

Value 0 - Specifies 1 | 0

Value 1 - Specifies 1 | 1 (One Lane in either direction)

" 2 - " " 1 | 2

" 3 - " " 1 | 3

" 4 - " " 2 | 2

" 5 - " " 2 | 3

" 6 - " " 3 | 3

```

        Bit 3-15 - NULL
        Bit 16 = 0 - N Logging
        Bit 16 = 1 - N+1 Logging
        Bit 17-31 - NULL"
 ::= { trafficCounterConfig 1 }

--Mod V4.01 - Expand to allow for 8 counters
trafficCounterSetDirection OBJECT-TYPE
    SYNTAX INTEGER (1..8)
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object sets the direction for flow and occupancy configuration and data values."
 ::= { trafficCounterConfig 2 }

--Mod V4.01 - Change from 5 minute to x minute values
trafficCounterFlowThresholdUp OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This Object sets the Up alarm threshold for x minute flow for the specified direction"
 ::= { trafficCounterConfig 3 }

trafficCounterFlowThresholdDown OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This Object sets the Down alarm threshold for x minute flow for the specified
direction"
 ::= { trafficCounterConfig 4 }

trafficCounterOccThresholdUp OBJECT-TYPE
    SYNTAX INTEGER(0..1000)-- Percentage Occupancy * 10
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object sets the Up alarm threshold for x minute occupancy for the specified
direction"
 ::= { trafficCounterConfig 5 }

trafficCounterOccThresholdDown OBJECT-TYPE
    SYNTAX INTEGER(0..1000) -- Percentage Occupancy * 10
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "This object sets the Down alarm threshold for x minute occupancy for the specified
direction"
 ::= { trafficCounterConfig 6 }

```

trafficCounterSpeedThresholdUp OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"This object sets the Up alarm threshold for x minute average speed for the specified direction (km/hr)"  
 ::= { trafficCounterConfig 7 }

trafficCounterSpeedThresholdDown OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"This object sets the Down alarm threshold for x minute average speed for the specified direction (km/hr)"  
 ::= { trafficCounterConfig 8 }

trafficCounterAlarmDirection OBJECT-TYPE  
SYNTAX INTEGER (1..8)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This Object returns the direction for an alarm"  
 ::= { trafficCounterConfig 9 }

trafficCounterIPAddress OBJECT-TYPE  
SYNTAX IpAddress  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
" This object hold the IP Address to which traps are returned.  
If the object is invalid or 0.0.0.0 then traps are returned to the IP Address of the manager which last made a Set or Get request"  
 ::= { trafficCounterConfig 10 }

trafficCounterPort OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
" This object hold the Port number to which traps are returned.  
If the object is invalid or 0 then traps are returned to the local Port of the manager which last made a Set or Get request"  
 ::= { trafficCounterConfig 11 }

trafficCounterQueueLoop OBJECT-TYPE  
SYNTAX INTEGER  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
" This object sets the detector for the Queue detection for the specified direction"

::= { trafficCounterConfig 12 }

trafficCounterQueueCallTime OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 " This object sets the time (in seconds) that the Queue detector  
 must be occupied before a queue alarm is raised"  
 ::= { trafficCounterConfig 13 }

trafficCounterQueueCancelTime OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 " This object sets the time (in seconds) that the Queue detector  
 must be un-occupied before a queue alarm is cancelled"  
 ::= { trafficCounterConfig 14 }

--Mod V4.01  
 trafficCounterAlarmInterval OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS mandatory  
 DESCRIPTION  
 " This object sets the time (in minutes) over which the threshold alarms operate."  
 ::= { trafficCounterConfig 15 }

--Mod V4.01 - Table of Detector inputs and road configuration  
 trafficCounterDetectorTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF TrafficCounterDetectorEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "Table contains detector and lane configuration"  
 ::= { trafficCounterConfig 16 }

trafficCounterDetectorEntry OBJECT-TYPE  
 SYNTAX TrafficCounterDetectorEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
 "This object is used for detector and lane configuration."  
 INDEX {trafficCounterDetectorCounter}  
 ::= {trafficCounterDetectorTable 1}

TrafficCounterDetectorEntry ::= SEQUENCE {  
     trafficCounterDetectorCounter INTEGER (1..8),  
     trafficCounterDetectorNoLanes INTEGER,  
     trafficCounterDetectorNPlusOne TruthValue,  
     trafficCounterDetectorDescription DisplayString

}

trafficCounterDetectorCounter OBJECT-TYPE

SYNTAX INTEGER (1..8)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Sets or returns the Direction/Counter number"

::={trafficCounterDetectorEntry 1}

trafficCounterDetectorNoLanes OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Sets or returns the number of lanes for the Direction/Counter"

::={trafficCounterDetectorEntry 2}

trafficCounterDetectorNPlusOne OBJECT-TYPE

SYNTAX TruthValue

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Sets or returns n (false) or n+1 (true) counting for this direction/counter"

::={trafficCounterDetectorEntry 3}

trafficCounterDetectorDescription OBJECT-TYPE

SYNTAX DisplayString

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Sets or returns the description for the Direction/Counter"

::={trafficCounterDetectorEntry 4}

--Mod V4.01 - Add TrapTrigger object

trafficCounterTrapTrigger OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS mandatory

DESCRIPTION

"This object sets which state to trigger a trap. A trap will trigger only once on each entry to trap configured

state

b0 (1) - Flow

b1 (2) - Occupancy

b2 (4) - Speed

b3 (8) - Queue

b4 (16)- Fault"

::= { trafficCounterConfig 17 }

-----  
-- Data

-----  
 --Mod V4.01 - Deprecate and use trafficCounterStartRTC

```
trafficCounterStartTime OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS deprecated
    DESCRIPTION
        "Sets the start time in seconds since 1/1/1970 00:00 for download of data"
    ::= { trafficCounterData 1 }
```

--Mod V4.01 - Setting a value which the traffic counter can not support returns a badValue error and leaves the

--value unchanged.

```
trafficCounterPeriod OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS mandatory
    DESCRIPTION
        "Sets the period in minutes for download of data.
        Only certain periods may be supported in which case the traffic counter
        will default to the last supported value and return a badValue error. As a minimum 5,
        15 & 60 minutes must be supported"
    ::= { trafficCounterData 2 }
```

```
trafficCounterFlow OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This object returns the total flow for the chosen time, period and direction"
    ::= { trafficCounterData 3 }
```

```
trafficCounterOccupancy OBJECT-TYPE
    SYNTAX INTEGER -- Occupancy (Percentage * 10)
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This object returns the total percentage occupancy for the chosen time, period and
        direction"
    ::= { trafficCounterData 4 }
```

--Mod V4.01 - Deprecate and replace with a table

```
trafficCounterClassification OBJECT-TYPE
    SYNTAX OCTET STRING(SIZE(32))
    -- INTEGER - Total flow for Class 1
    -- INTEGER - Total flow for Class 2
    -- INTEGER - Total flow for Class 3
    -- INTEGER - Total flow for Class 4
    -- INTEGER - Total flow for Class 5
    -- INTEGER - Total flow for Class 6
    -- INTEGER - Total flow for Class 7
    -- INTEGER - Total flow for Class 8
```

ACCESS read-only  
 STATUS deprecated  
 DESCRIPTION  
 "This object returns the flow by classification for the chosen time, period and direction"  
 ::= { trafficCounterData 5 }

--Mod V4.01 - Speed in km/hr multiplied by 10  
 trafficCounterSpeed OBJECT-TYPE  
 SYNTAX INTEGER -- km/hr \* 10  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
 "This object returns the average speed (km/hr \* 10) for the chosen time, period and direction"  
 ::= { trafficCounterData 6 }

--Mod V4.01 - Change to return the percentage of time the queue was present over the period specified  
 trafficCounterQueue OBJECT-TYPE  
 SYNTAX INTEGER (0..1000) -- Percent \* 10  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
 "This object returns the percentage of time that the Queue existed for the chosen time, period and direction"  
 ::= { trafficCounterData 7 }

--Mod V4.01 - Change to return the percentage time the queue was present  
 trafficCounterHeadway OBJECT-TYPE  
 SYNTAX INTEGER -- Average in seconds \* 10  
 ACCESS read-only  
 STATUS optional  
 DESCRIPTION  
 "This object returns the average headway in seconds \* 10 for the chosen time, period and direction"  
 ::= { trafficCounterData 8 }

--Mod V4.01 - Add table for classification  
 trafficCounterClassificationTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF TrafficCounterClassificationEntry  
 ACCESS not-accessible  
 STATUS optional  
 DESCRIPTION  
 "Table contains classification data"  
 ::= { trafficCounterData 9 }

trafficCounterClassificationEntry OBJECT-TYPE  
 SYNTAX TrafficCounterClassificationEntry  
 ACCESS not-accessible  
 STATUS optional  
 DESCRIPTION  
 "This object is used for classification data."  
 INDEX {trafficCounterClassificationClass}

```
 ::= {trafficCounterClassificationTable 1}
```

```
TrafficCounterClassificationEntry ::= SEQUENCE {
    trafficCounterClassificationClass    INTEGER,
    trafficCounterClassificationFlow    INTEGER,
    trafficCounterClassificationSpeed   INTEGER,
    trafficCounterClassificationDesc    DisplayString
}
```

```
trafficCounterClassificationClass    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Classification class number"
    ::= {trafficCounterClassificationEntry 1}
```

```
trafficCounterClassificationFlow    OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Classification flow for the chosen start time, period, direction and class"
    ::= {trafficCounterClassificationEntry 2}
```

```
trafficCounterClassificationSpeed    OBJECT-TYPE
    SYNTAX INTEGER          -- Speed in km/hr * 10
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Classification speed (km/hr * 10) for the chosen start time, period, direction and
class"
    ::= {trafficCounterClassificationEntry 3}
```

```
trafficCounterClassificationDesc    OBJECT-TYPE
    SYNTAX DisplayString
    ACCESS read-write
    STATUS optional
    DESCRIPTION
        "Classification description"
    ::= {trafficCounterClassificationEntry 4}
```

```
-----
-- Faults
-----
```

```
trafficCounterFaultTable    OBJECT-TYPE
    SYNTAX SEQUENCE OF TrafficCounterFaultEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
```

"Provides a table of instances of fault data."  
 ::= { trafficCounterFault 1 }

trafficCounterFaultEntry OBJECT-TYPE  
 SYNTAX TrafficCounterFaultEntry  
 ACCESS not-accessible  
 STATUS mandatory  
 DESCRIPTION  
   "A list of faults"  
 INDEX { trafficCounterFaultNo }  
 ::= { trafficCounterFaultTable 1 }

TrafficCounterFaultEntry ::=  
 SEQUENCE {  
   trafficCounterFaultNo INTEGER,  
   trafficCounterFaultID INTEGER,  
   trafficCounterFaultType INTEGER,  
   trafficCounterFaultDate INTEGER,  
   trafficCounterFaultDirection INTEGER,  
   trafficCounterFaultRTC UTMCTime  
 }

trafficCounterFaultNo OBJECT-TYPE  
 SYNTAX INTEGER (0..100)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
   >Returns the index for a fault entry"  
 ::= { trafficCounterFaultEntry 1 }

trafficCounterFaultID OBJECT-TYPE  
 SYNTAX INTEGER {  
   mainsFail(1),  
   memoryFull(2),  
   detectorFail(10),  
   permanentDetection(20),  
   noDetection(30),  
   commsFailToDetector(40)  
 }  
  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
   >Returns the identifier for the fault type"  
 ::= { trafficCounterFaultEntry 2 }

trafficCounterFaultType OBJECT-TYPE  
 SYNTAX INTEGER {  
   equipment(1),  
   comms(2)  
 }  
 ACCESS read-only

STATUS deprecated  
 DESCRIPTION  
 "Returns the sub-type for the fault type"  
 ::= { trafficCounterFaultEntry 3 }

trafficCounterFaultDate OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-only  
 STATUS deprecated  
 DESCRIPTION  
 "Returns the time & date for the fault in seconds since 1st Jan 1970."  
 ::= { trafficCounterFaultEntry 4 }

--Mod V4.01 - Allow up to 8 counters  
 trafficCounterFaultDirection OBJECT-TYPE  
 SYNTAX INTEGER (1..8)  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "Returns the direction for the alarm"  
 ::= { trafficCounterFaultEntry 5 }

--Mod V4.01 - Add RTC object to read date  
 trafficCounterFaultRTC OBJECT-TYPE  
 SYNTAX UTMCTime  
 ACCESS read-only  
 STATUS mandatory  
 DESCRIPTION  
 "Returns the direction for the alarm"  
 ::= { trafficCounterFaultEntry 6 }

-----  
 -- Dual Loop Configuration  
 -----

--Mod V4.01 - Deprecate as these configuration objects are equipment/manufacturer specific

trafficCounterDualCount OBJECT-TYPE  
 SYNTAX DisplayString (SIZE(5))  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
 "Used to read and write the configuration settings for dual loop counting:-  
 Format Char1Char2Char3Char4Char5  
 Char 1 ' 1' = all single loop counting  
 ' 2' = 1 dual loop count 6 single loop  
 ' 3' = 2 dual loop counts 4 single loop  
 ' 4' = 3 dual loop counts 2 single loop  
 ' 5' = 4 dual loop counts 0 single loop  
 Chars 2 - 5 ' 1' = bi-directional count (Char 2 applies to first dual counter)  
 ' 0' = uni-directional count"  
 ::= { trafficCounterDual 1 }

trafficCounterDualTable OBJECT-TYPE  
 SYNTAX SEQUENCE OF TrafficCounterDualEntry  
 ACCESS not-accessible  
 STATUS deprecated  
 DESCRIPTION  
 "Table contains unit configuration with respect to monitored counters."  
 ::= { trafficCounterDual 2 }

trafficCounterDualEntry OBJECT-TYPE  
 SYNTAX TrafficCounterDualEntry  
 ACCESS not-accessible  
 STATUS deprecated  
 DESCRIPTION  
 "This object is used for configuration values relating to a counter."  
 INDEX { trafficCounterDualNum }  
 ::= { trafficCounterDualTable 1 }

TrafficCounterDualEntry ::= SEQUENCE {  
     trafficCounterDualNum INTEGER,  
     trafficCounterDualDirection INTEGER,  
     trafficCounterDualDistance INTEGER,  
     trafficCounterStartLoop INTEGER,  
     trafficCounterLoopType INTEGER  
 }

trafficCounterDualNum OBJECT-TYPE  
 SYNTAX INTEGER (1..64)  
 ACCESS read-only  
 STATUS deprecated  
 DESCRIPTION  
 "counter number (1->N)"  
 ::= { trafficCounterDualEntry 1 }

trafficCounterDualDirection OBJECT-TYPE  
 SYNTAX INTEGER {  
     dir1(1),  
     dir2(2),  
     uNASSIGNED(3)  
 }  
 ACCESS read-write  
 STATUS deprecated  
 DESCRIPTION  
 "States if the counter is a direction 1 or direction 2 counter."  
 ::= { trafficCounterDualEntry 2 }

trafficCounterDualDistance OBJECT-TYPE  
 SYNTAX INTEGER  
 ACCESS read-write  
 STATUS deprecated

DESCRIPTION

"The distance in mm between dual loops or the detection distance for a single loop.

Value = 0 - Standard 10ft (3050mm) loop

Value = 1 - Standard 12ft (3660mm) loop

Value = 2..9 predefined

Value >= 10 - distance in mm"

::= { trafficCounterDualEntry 3 }

trafficCounterStartLoop OBJECT-TYPE

SYNTAX INTEGER

ACCESS read-write

STATUS deprecated

DESCRIPTION

"The loop number of the first loop connected to this counter."

::= { trafficCounterDualEntry 4 }

trafficCounterLoopType OBJECT-TYPE

SYNTAX INTEGER {

single(1),

unidirectional(2),

bidirectional(3)

}

ACCESS read-only

STATUS deprecated

DESCRIPTION

"The number of counters on a unit."

::= { trafficCounterDualEntry 5 }

END

## **F Definitions of registered IDL scripts (Normative)**

- F.1 IDL scripts for use in UTMC systems are described in the spreadsheet “TS004 Annex F Mar05” that accompanies this document.
- F.2 These standardised scripts should be used wherever possible to manage the data passed across a CORBA link within a UTMC system.

## **G Definitions of registered Other Objects (Normative)**

G.1 There are no other objects currently specified.