



UTMC-TS004.005:2009

UTMC Objects Registry

10 December 2009

Cover + 149 pages

© Copyright 2009

Foreword

This document, UTMC Technical Specification 004 version 005 (TS004.005:2009), was prepared by the UTMC Technical Secretary with the support of the UTMC Development Group and the Department for Transport (DfT). It forms part of the range of UTMC specifications and supersedes previous versions of the Objects Registry.

TS004 presents the data standards recommended for use by UK traffic managers in their systems. Details are provided in normative annexes.

TS004.005:2009 is a revision of the earlier document TS004.004:2008, published in December 2008. The changes made have been in response to feedback from users and suppliers to improve the scope, accuracy, consistency and utility of the previous version, in accordance with the published consultation process.

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 intended to be stable for several years at a time.

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

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 specifications, such specifications are not mandatory.

We gratefully acknowledge the considerable amount of work contributed to the development of this specification, and of products that conform to it, by the UTMC systems industry.

List of contents

Foreword	1
1 Introduction	3
1.1 General	3
1.2 Intellectual Property Rights and usage of this document	3
1.3 Document approach and structure	3
2 Object registration	5
2.1 Object reference	5
2.2 Registration of an Object	5
A References (Normative)	6
B Management authority for this document (Informative)	7
B.1 Management authority	7
B.2 National authority	7
C Schedule of registered objects (Informative)	8
C.1 Registered Objects	8
D Definitions of registered Data Objects (Normative)	13
D.1 UTMC UML model	13
D.2 Residual spreadsheet model	13
E Definitions of registered MIBs (Normative)	14
E.1 Introduction	14
E.2 UM/001, UTMC header MIB	14
E.3 UM/002, Air quality monitor MIB	17
E.4 UM/003, VMS MIB	32
E.5 UM/004, Simple UTC MIB	44
E.6 UM/005, UTC MIB	50
E.7 UM/006, Car Park Monitor MIB	70
E.8 UM/007, Traffic Counter MIB	85
E.9 UM/008, Full UTC MIB	100
F Definitions of registered IDL scripts (Normative)	134
F.1 Introduction	134
F.2 UI/001, BCD.idl	134
F.3 UI/002, MJD.idl	135
F.4 UI/003, B-Query.idl	136
F.5 UI/004, B-SessionManagement.idl	138
F.6 UI/005, B-Subscriptions.idl	140
F.7 UI/006, B-TabularResults.idl	143
F.8 UI/007, B-Utility.idl	146
G Definitions of registered XML Schemas (Normative)	148
G.1 Introduction	148
H Definitions of registered Other Objects (Normative)	149

1 Introduction

1.1 General

- 1.1.1 TS004 ¹ 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 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 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.

1.2 Intellectual Property Rights and usage of this document

- 1.2.1 Intellectual Property Rights in the UTMC Technical Specification, including the Objects specified in this document, are protected. Those proposing changes shall be deemed to acknowledge and accept this, and to accept responsibility for ensuring that there are no third party IPR claims on the proposed change.
- 1.2.2 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.
- 1.2.3 This document is designed for use alongside TS003, the UTMC Technical Specification, which provides a framework of general standards for UTMC systems.

1.3 Document approach and structure

- 1.3.1 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.
- 1.3.2 The following annexes are included. Annexes A, D, E, F, G and H 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.

¹ In this volume, TS004 means TS004.005:2009 unless the contrary is specified

- 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 XML objects: a series of XSD schemas for use in UTMC systems exchanging information over XML/HTTP connections.
- Annex H: Definitions of registered Other Objects: the specification of objects to be used under circumstances other than those in Annexes D, E, F and G. There are currently no entries in this Annex.

1.3.3 Annexes D.1, D.2 and G are presented in separate documents: Annex D.1 is large, and Annexes D.2 and G are in different formats (D.2 is a spreadsheet, G is a zipped folder).

2 Object registration

2.1 Object reference

- 2.1.1 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 with nnn denoting the reference number allocated to the Object by the Registry. The exception is for XML objects where this numbering structure is unhelpful, and package names may be used where this degree of specificity is required.
- 2.1.2 Entries in the UTMC Product Catalogue should cite specific Objects wherever they are applicable.
- 2.1.3 Users procuring products, systems or services for integration into an existing UTMC system should cite the Objects that are currently in use, and those that are required in the new system.

2.2 Registration of an Object

- 2.2.1 The UDG operates a published procedure for the registration of a new or changed Object, which is available as UTMCD029 ("Procedure for submission of updates to the UTMC Technical Specification"). This procedure adheres to the following principles:
- a) *Notification.* A Proposer may at any time notify the UDG, via the Technical Secretary, of its intention to develop and register an Object. Objects should be *necessary*: duplicate or alternative Objects should be avoided.
 - b) *Submission.* Once the Object has been developed, the Proposer submits it to the UDG, via the Technical Secretary. Objects should respect the goal of backwards compatibility between Technical Specification versions.
 - c) *Consultation.* Once the UDG is satisfied that the proposed Object has been scrutinised sufficiently, it will be included in a Consultation Draft for an upissue of the Technical Specification. Consultation on TS004 upissues will follow current UK Government guidance as far as is practicable.
 - d) *Adoption.* If, on the closure of public consultation, there have been no significant and substantive comments, the proposed Object will be adopted in a formal upissue of the Technical Specification. An announcement will be made of the upissue.
 - e) *Retirement.* An Object may be deprecated, or even withdrawn from, an upissue of the Technical Specification as a result of a proposed change. As this step is in principle a breach of the backwards compatibility principle, it will require public consultation in the same way that adoption does.

A References (Normative)

- A.1 The Objects Registry represents semantic structures for use within UTMC systems. The normative requirements on systems employing these Objects are described within the Framework Technical Specification, TS003.

B Management authority for this document (Informative)

B.1 Management authority

B.1.1 The Technical Specification is formally and managed by UTMC Ltd on behalf of UTMC Development Group (UDG), a cooperative grouping of local authorities and system suppliers. The UDG is currently the “relevant body” for matters relating to UTMC compliance.

B.1.2 Under delegated authority the UDG Specifications and Standards Group oversees developments of this and other UTMC technical documentation and procedures on a day to day basis.

B.1.3 The contact address for this specification, from which the current issue of the Technical Specification and advice on its use can be obtained, is as follows:

UTMC Technical Secretary
UTMC Ltd
Surrey Technology Centre
Surrey Research Park
Guildford
Surrey GU2 7YG
United Kingdom

Tel: +44 (0) 1483 688270
Fax: +44 (0) 1483 688271
E-mail: secretariat@utmc.uk.com

B.1.4 Any changes to this will be published on the UTMC website www.utmc.uk.com.

B.2 National authority

B.2.1 The Highways Agency, an Agency of the Department for Transport, sponsors the UTMC Technical Specification. The contact point within the Agency is:

Azra Zohrabi
Highways Agency
Federated House
London Road
Dorking
Surrey RH4 1SZ
United Kingdom

B.2.2 The UK Department for Transport endorses the use of the UTMC Technical Specification. The contact point within DfT Centre is as follows:

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

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.

Data Objects (see Annex D)

C.1.2 NB the majority of these Objects are now available as UML specifications and may be more suitably referred to by package/class names; see Annex D.1 (separate document).

Reference	Name	Current version/TS004 entry
UC/001	Access Control Data Object	27/4/07
UC/002	Accident Data Object	27/4/07
UC/003	Air Quality Data Object	27/4/07
UC/004	Car Park Data Object	27/4/07
UC/005	CCTV Data Object	27/4/07
UC/006	Common Support Types Data Objects <i>Quality Support Object</i> <i>Device History Support Object</i> <i>Command Support Object</i>	27/4/07
UC/007	Common Subsystem Support Object	27/4/07
UC/008	Common TypeID Support Object	27/4/07
UC/009	DayType Support Object	27/4/07
UC/010	Detector Data Object	27/4/07
UC/011	Event Data Object	27/4/07
UC/012	Fault Support Objects <i>Fault Support Object</i> <i>Acknowledgement State Support Object</i> <i>Fault Type Support Object</i>	27/4/07
UC/013	Global Support Object	27/4/07
UC/014	Incident Data Object	27/4/07
UC/015	Meteorological Data Object	27/4/07
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>	27/4/07
UC/017	Prediction Data Object	27/4/07
UC/018	Profile Data Object	27/4/07
UC/019	Roadworks Data Object	27/4/07
UC/020	Traffic Signal Data Object	27/4/07

C Schedule of registered objects (Informative)

Reference	Name	Current version/TS004 entry
UC/021	Transport Link Data Object	27/4/07
UC/022	Transport Route Data Object	27/4/07
UC/023	VMS Data Object <i>VMS Messages Support Object</i> <i>VMS Message List Support Object</i> <i>VMS Car Park List Support Object</i>	27/4/07
UC/024	ANPR Data Object	30/9/09

MIBs (see Annex E)

Reference	Name	Current version	TS004 entry
UM/001	UTMC Header MIB	1.05	30/3/05
UM/002	Air Quality Monitor MIB	2.01	30/3/05
UM/003	VMS MIB	3.01	30/3/05
UM/004	Simple UTC MIB – SUPERSEDED BY UM/008	0.2	30/3/05
UM/005	UTC MIB – SUPERSEDED BY UM/008	1.12	30/3/05
UM/006	Car Park Monitor MIB	2.01	30/3/05
UM/007	Traffic Counter MIB	4.01	30/3/05
UM/008	Full UTC MIB	1.0	19/8/08

C.1.3 A number of MIBs have associated guidance documents, which are available separately.

IDL scripts (see Annex F)

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

XML schemas (see Annex G)

C.1.4 A schema is now available covering the full range of UTMC registered Data Objects, as well as some additional functions including services on alerts and interventions developed under the Government supported project FREEFLOW. It builds on and incorporates transport mechanisms of the European protocols SIRI and DATEX II.

C *Schedule of registered objects (Informative)*

C.1.5 Annex G is made available as a zipped file containing a series of XSD files, and a small number of other informative files. This file is normative.

C.1.6 The packages included are as follows.

Name	Current version/TS004 entry
(top level)	
utmc_publication.xsd	30/9/09
utmc_siri.xsd	30/9/09
utmcX.xsd	30/9/09
utmc folder	
utmc_all_devices-v1.0.xsd	30/9/09
utmc_all_objects-v1.0.xsd	30/9/09
utmc_all_quantiseds-v1.0.xsd	30/9/09
utmc_all_traffic_events-v1.0.xsd	30/9/09
utmc_data_object_access_control-v1.0.xsd	30/9/09
utmc_data_object_air_quality-v1.0.xsd	30/9/09
utmc_data_object_car_park-v1.0.xsd	30/9/09
utmc_data_object_cctv-v1.0.xsd	30/9/09
utmc_data_object_detector-v1.0.xsd	30/9/09
utmc_data_object_meteorological-v1.0.xsd	30/9/09
utmc_data_object_traffic_signal-v1.0.xsd	30/9/09
utmc_data_object_transport_link-v1.0.xsd	30/9/09
utmc_data_object_transport_route-v1.0.xsd	30/9/09
utmc_device_command-v1.0.xsd	30/9/09
utmc_device_fault-v1.0.xsd	30/9/09
utmc_device_history-v1.0.xsd	30/9/09
utmc_device-v1.0.xsd	30/9/09
utmc_location-v1.0.xsd	30/9/09
utmc_network_base-v1.0.xsd	30/9/09
utmc_network-v1.0.xsd	30/9/09
utmc_object-v1.0.xsd	30/9/09
utmc_quantised_access_control-v1.0.xsd	30/9/09
utmc_quantised_air_quality-v1.0.xsd	30/9/09
utmc_quantised_car_park-v1.0.xsd	30/9/09
utmc_quantised_cctv-v1.0.xsd	30/9/09
utmc_quantised_detector-v1.0.xsd	30/9/09
utmc_quantised_traffic_signal-v1.0.xsd	30/9/09

C Schedule of registered objects (Informative)

Name	Current version/TS004 entry
utmc_quantised_transport_link-v1.0.xsd	30/9/09
utmc_quantised_transport_route-v1.0.xsd	30/9/09
utmc_quantised-v1.0.xsd	30/9/09
utmc_support_data_source-v1.0.xsd	30/9/09
utmc_support_day_type-v1.0.xsd	30/9/09
utmc_support_meteorological-v1.0.xsd	30/9/09
utmc_support_quality-v1.0.xsd	30/9/09
utmc_support_subsystem-v1.0.xsd	30/9/09
utmc_support_typeID-v1.0.xsd	30/9/09
utmc_traffic_event_accident-v1.0.xsd	30/9/09
utmc_traffic_event_event-v1.0.xsd	30/9/09
utmc_traffic_event_incident-v1.0.xsd	30/9/09
utmc_traffic_event_roadworks-v1.0.xsd	30/9/09
utmc_traffic_event-v1.0.xsd	30/9/09
utmc_utility_time-v1.0.xsd	30/9/09
utmc_utility-v1.0.xsd	30/9/09
<i>utmc_anpr folder</i>	
ANPRDiagnostic.xsd	30/9/09
Camera.xsd	30/9/09
CameraToInstation.xsd	30/9/09
Config.xsd	30/9/09
Image.xsd	30/9/09
InstationToCamera.xsd	30/9/09
Lane.xsd	30/9/09
MachineDiagnostic.xsd	30/9/09
Plateread.xsd	30/9/09
types.xsd	30/9/09
<i>This folder also includes the following WSDL files</i>	
camera.wsdl	30/9/09
instation.wsdl	30/9/09
<i>utmc_freelflow folder</i>	
utmc_all-v1.0.xsd	30/9/09
utmc_data_object_intervention-v1.0.xsd	30/9/09
utmc_data_object_transport_route_diversion-v1.0.xsd	30/9/09
utmc_filter_intervention-v1.0.xsd	30/9/09
utmc_filter_traffic_alert-v1.0.xsd	30/9/09

C Schedule of registered objects (Informative)

Name	Current version/TS004 entry
utmc_traffic_event_alert-v1.0.xsd	30/9/09
utmcsvc folder	
utmc_all-v1.0.xsd	30/9/09
utmc_all_filters-v1.0.xsd	30/9/09
utmc_filter_device-v1.0.xsd	30/9/09
utmc_filter_network-v1.0.xsd	30/9/09
utmc_filter_object-v1.0.xsd	30/9/09
utmc_filter_prediction-v1.0.xsd	30/9/09
utmc_filter_profile-v1.0.xsd	30/9/09
utmc_filter_quantised-v1.0.xsd	30/9/09
utmc_filter_traffic_event-v1.0.xsd	30/9/09
utmc_request_service-v1.0.xsd	30/9/09
siri folder	
siri_all.xsd	30/9/09
siri_base-v1.3.xsd	30/9/09
siri_common-v1.3.xsd	30/9/09
siri_location-v1.1.xsd	30/9/09
siri_modes-v1.1.xsd	30/9/09
siri_participant-v1.1.xsd	30/9/09
siri_reference-v1.2.xsd	30/9/09
siri_requests-v1.2.xsd	30/9/09
siri_situationActions-v1.0.xsd	30/9/09
siri_time-v1.2.xsd	30/9/09
siri_types-v1.1.xsd	30/9/09
siri_utility-v1.1.xsd	30/9/09
datex2 folder	
DatexII_publication.xsd	30/9/09
DATEXII Schema_1_0_1_0.xsd	30/9/09
DATEXII Schema_1_0_1_0_subset.xsd	30/9/09

Other objects (see Annex H)

C.1.7 There are currently no other registered Objects.

D Definitions of registered Data Objects (Normative)

D.1 UTMC UML model

- D.1.1 In TS004.004:2008 the main body of UMTC Data Objects was re-rendered into UML, making it better specified and adding some additional clarity. Annex D.1 contains this model presented in the form of package/class tables with some diagrams for additional clarity. This is a large document and is published separately for ease of use.
- D.1.2 These standardised Data Objects should be used wherever possible to structure the data passed across an open interface within a UTMC system.
- D.1.3 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.

D.2 Residual spreadsheet model

- D.2.1 Included for the first time in TS004.005:2009 is a model for ANPR. Because of the timescale over which this has been developed, it has not been incorporated into the UML model and retains the presentation form of previous Technical Specification issues, namely a series of connected spreadsheets. These are published separately as Annex D.2.

E Definitions of registered MIBs (Normative)

E.1 Introduction

E.1.1 MIBs for use in UTMC systems are presented in text form below.

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/001, 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 }

```

E Definitions of registered MIBs (Normative)

```
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/002, 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 UTMC compliant outstations.

-- Copyright City of York Council 2002
```

IMPORTS

```
TRAP-TYPE
    FROM RFC-1215
OBJECT-TYPE
    FROM RFC-1212
    utmc, utmcAirQualType1, UTMCTime, DisplayString
        FROM UTMC-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

supported

"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

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/003, 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/004, 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/005, 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 }

sSOSStatus 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 Gauge32,
cfREPbit Integer32,
dfREPbit Integer32,
wiREPbit Integer32,
rrREPbit Integer32,
loREPbit Integer32,

```

```

sfREPbit Integer32,
lf1REPbit Integer32,
mmREPbit Integer32,
lf2REPbit Integer32,
vsREPbits Integer32,
sdREPbytes OCTET STRING,
vcREPbit Integer32,
odREPbit Integer32,
qdREPbit Integer32,
gwREPCancel Integer32,
gwREPdetector Integer32,
gwREPrequest Gauge32,
udsfREPbit Integer32,
beREPrunning Integer32,
lrtREPmode Integer32,
lrtREPdetectorsuspect
Integer32,
lrtREPlampfail Integer32,
lrtREPdetectorfail
Integer32,
lrtREPwatchdogfail
Integer32,
lrtREPphasereply
Gauge32,
dREPbits Gauge32,
tsREPbit Integer32,
doREPbit Integer32,
ehREPbit Integer32,
evREPbit Integer32,
vgREPbit Integer32,
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/006, 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/007, 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/manufacture 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

E.9 UM/008, Full UTC MIB

```

--
-- UTMC-UTMCFULLUTCTYPE2-MIB-1rc2.my
-- Version 1.0
-- MIB generated by MG-SOFT Visual MIB Builder Version 6.0 Build 88
-- Monday, May 19, 2008 at 09:42:31
--
    UTMC-UTMCFULLUTCTYPE2-MIB DEFINITIONS ::= BEGIN
        IMPORTS
            OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
            FROM SNMPv2-CONF
            enterprises, Integer32, OBJECT-TYPE, MODULE-IDENTITY,
NOTIFICATION-TYPE
            FROM SNMPv2-SMI
            TEXTUAL-CONVENTION
            FROM SNMPv2-TC;
-- 1.3.6.1.4.1.13267.3.2
utmcFullUTCType2 MODULE-IDENTITY
    LAST-UPDATED "200710161421Z" -- October 16, 2008 at 14:21 GMT
    ORGANIZATION
        "Peek Traffic Ltd & Siemens Traffic Controls "
    CONTACT-INFO
        "UTMC Group
        http://utmc.uk.com/
        secretariat@utmc.uk.com"
    DESCRIPTION
        "UTC Core MIB for UTC Control"
    REVISION "200805190941Z" -- May 19, 2008 at 09:41 GMT
    DESCRIPTION
        "Added Correct Contact
        details for UTMC group. "
    REVISION "200805190938Z" -- May 19, 2008 at 09:38 GMT
    DESCRIPTION
        "Added utcType2OperationMode
        description for different
        modes."
    REVISION "200804301625Z" -- April 30, 2008 at 16:25 GMT
    DESCRIPTION
        "Added MOVA Objects
        Added Time object
        Spell checks
        Added Re-send holdoff
        Re-Worded Some Text descriptions
        Renamed Max Interval to keep alive.
        "
    REVISION "200804161617Z" -- April 16, 2008 at 16:17 GMT
    DESCRIPTION
        "Added missing objects
        agreed by Peek and Siemens

```

to be common but not in
TR spec."

REVISION "200804151431Z" -- April 15, 2008 at 14:31 GMT
DESCRIPTION
"added utcType2ReplyByExceptionMaxInterval
to allow for a heartbeat"

REVISION "200803191557Z" -- March 19, 2008 at 15:57 GMT
DESCRIPTION
"added changes from meeting 12 Feb 2008
between peek and siemens"

REVISION "200802041438Z" -- February 04, 2008 at 14:38 GMT
DESCRIPTION
"Changed scoot Description to
match old description.
Updated description
for reply and control
time stamps.
Updated reply by
exception description.
"

REVISION "200802041017Z" -- February 04, 2008 at 10:17 GMT
DESCRIPTION
"Updated the description
for bit mask."

REVISION "200802040932Z" -- February 04, 2008 at 09:32 GMT
DESCRIPTION
"Made ReplyByExceptionRetryDelay
be milliseconds.
Added type for Site ID
Added type for Timestamp
Reordered Site and timestamp
"

REVISION "200801240953Z" -- January 24, 2008 at 09:53 GMT
DESCRIPTION
"Moved Scoot detector
count to status tree
because it is read only."

REVISION "200801151609Z" -- January 15, 2008 at 16:09 GMT
DESCRIPTION
"Added GP and VO oids.
Added timestamp"

REVISION "200711071221Z" -- November 07, 2007 at 12:21 GMT
DESCRIPTION
"Removed Seconds of day
from reply table.
Added Seconds of Day
To notification"

REVISION "200710190818Z" -- October 19, 2007 at 08:18 GMT
DESCRIPTION
"Moved Top OID to utmc
under utmcFullUTCType2"

REVISION "200710181007Z" -- October 18, 2007 at 10:07 GMT

```

        DESCRIPTION
            "Beta Ready For Review"
        ::= { utmcFullUTC 2 }
--
-- Textual conventions
--
UTCType2TruthValue ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "Represents a boolean value."
    SYNTAX INTEGER
        {
            false(0),
            true(1)
        }
UTCType2BitMask ::= TEXTUAL-CONVENTION
    DISPLAY-HINT
        "1x "
    STATUS current
    DESCRIPTION
        "A bit pattern represented as a byte array where the first byte
        contains the first 8 bits in least significant bit order,
        2nd byte contains the next 8 bits, &c. Each bit performs a
        similar function, e.g. stage force bits, the l.s.b is Force-1
        then Force-2 etc. As most controllers don't have more than 8
        stages per stream this will rarely be more than 1 Octet"
    SYNTAX OCTET STRING (SIZE (1..8))
UTCType2SiteID ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "Identifies which, of possibly several, equipment at this site the
        object should be applied to. The format is a Free Text ASCII String
        This could be an SCN, IP address, or a number."
    SYNTAX OCTET STRING (SIZE (1..16))
UTCType2TimeStamp ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "Seconds of Day +2. Universal Time Co-ordinated Time.
        A timestamp value of 1 should be taken to mean 'NOW'.
        e.g. Midnight = +2
        NOTE: 0 is not valid as SNMP indexing constraints do not allow
        this as an index"
    SYNTAX INTEGER (1..86401)
--
-- Node definitions
--
-- 1.3.6.1.4.1.13267
utmc OBJECT IDENTIFIER ::= { enterprises 13267 }
-- 1.3.6.1.4.1.13267.3
utmcFullUTC OBJECT IDENTIFIER ::= { utmc 3 }
-- 1.3.6.1.4.1.13267.3.2.1
utCType2Version OBJECT IDENTIFIER ::= { utmcFullUTCType2 1 }

```

```

-- 1.3.6.1.4.1.13267.3.2.1.1
utcType2MIBVersion OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "String for MIB Version.
        Format should be <major version>,<minor version> e.g. '1.2'
        "
    ::= { utcType2Version 1 }
-- 1.3.6.1.4.1.13267.3.2.1.2
utcType2AppVersion OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "String For Application Version.
        Format is vendor specific."
    ::= { utcType2Version 2 }
-- 1.3.6.1.4.1.13267.3.2.1.3
utcType2AppPartNumber OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Part number for the UTC application.
        Format is vendor specific."
    ::= { utcType2Version 3 }
-- 1.3.6.1.4.1.13267.3.2.1.4
utcType2VendorID OBJECT-TYPE
    SYNTAX OCTET STRING
    UNITS "Vendor"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object defines the Vendor. Example values being 'Peek'
        and 'Siemens', these will be fixed for these vendors."
    ::= { utcType2Version 4 }
-- 1.3.6.1.4.1.13267.3.2.1.5
utcType2HardwareType OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This Object defines the hardware platform name.
        Example Values will be 'Chameleon' and 'Gemini'.
        These will be fixed for these hardware platforms"
    ::= { utcType2Version 5 }
-- 1.3.6.1.4.1.13267.3.2.1.6
utcType2HardwareID OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION
    "This Object is vendor specific. Typically, this will start with
    vendor specific Hardware Serial Number, followed by optional
    vendor specific keywords."
    ::= { utcType2Version 6 }
-- 1.3.6.1.4.1.13267.3.2.2
utcType2Config OBJECT IDENTIFIER ::= { utmcFullUTCType2 2 }
-- 1.3.6.1.4.1.13267.3.2.2.1
utcType2ConfigLastChanged OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (15))
    UNITS "YYYYMMDDHHmmssZ"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object returns the time the configuration
        data (utcCoreConfig) available via the MIB was
        last changed, as YYYYMMDDHHmmssZ.
        Z indicates zulu or GMT
        This format is based upon ExtUTCTime from RFC:2578 and is the
        format used in other SNMP and UTMC timestamps.
        "
    ::= { utcType2Config 1 }
-- 1.3.6.1.4.1.13267.3.2.2.2
utcType2InstationAddress OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "SNMP TRAP IP address to use for Reply by Exception. This IP
        address will also be used for the default NTP server unless
        overridden by configuration data. The Type is a String so that a
        DNS entry can be used instead of an IP address.
        "
    ::= { utcType2Config 2 }
-- 1.3.6.1.4.1.13267.3.2.2.3
utcType2InstationPort OBJECT-TYPE
    SYNTAX Integer32 (1..65535)
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "SNMP TRAP Port number to use for Reply by Exception. "
    ::= { utcType2Config 3 }
-- 1.3.6.1.4.1.13267.3.2.2.4
utcType2OperationModeTimeout OBJECT-TYPE
    SYNTAX Integer32 (0..180)
    UNITS "Seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is the number of seconds that Operation Mode stays valid.
        If the Operation Mode has not been refreshed within this interval

```

```

the opMode drops to Standalone. Default Value = 60 seconds.
Setting this value to zero disables the timer.
"
 ::= { utcType2Config 4 }
-- 1.3.6.1.4.1.13267.3.2.2.5
utcType2ScootSampleReportInterval OBJECT-TYPE
    SYNTAX Integer32 (1..16)
    UNITS "Seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This is the number of seconds that SCOOT detector data
        is held before it gets transmitted. On non-mc3 systems this
        will be 1 second. On mc3 systems this will normally be 4
        seconds."
    ::= { utcType2Config 5 }
-- 1.3.6.1.4.1.13267.3.2.2.6
utcType2ReplyByException OBJECT-TYPE
    SYNTAX INTEGER
        {
            disable(0),
            enable(1)
        }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Enable or Disable the Reply By exception system.
        Default is disabled (0)"
    ::= { utcType2Config 6 }
-- 1.3.6.1.4.1.13267.3.2.2.7
utcType2ReplyByExceptionRetryDelay OBJECT-TYPE
    SYNTAX Integer32 (1..10000)
    UNITS "Milliseconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This object defines the number of milliseconds the OTU
        will wait for an acknowledge from the instation
        before retrying and sending the notification again.
        If the OTU can't support the specified value then it
        should round it up to the next value it can.
        Typical value would be 200ms for wired communications,
        lower values could be used for fibre and higher values
        could be used for wireless.
        A value of zero is invalid
        The default value will be 200ms."
    ::= { utcType2Config 7 }
-- 1.3.6.1.4.1.13267.3.2.2.8
utcType2ReplyByExceptionRetryCount OBJECT-TYPE
    SYNTAX Integer32 (0..100)
    UNITS "Number"

```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "This object defines the number of times the OTU
    will resend a notification to the instation
    before giving up. 0 will cause the OTU
    to not resend any failed notifications.
    The typically value should be 4.
    The default value will be 4.
    "
    ::= { utcType2Config 8 }
-- 1.3.6.1.4.1.13267.3.2.2.9
utcType2ReplyByExceptionKeepAlive OBJECT-TYPE
    SYNTAX Integer32 (0..30)
    UNITS "Seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This MIB object specifies the maximum time that's
        allow to pass between the OTU sending INFORM
        requests to the instation.
        0 would mean there is no maximum time; the OTU only sends
        informs when the reply data changes.
        Other values require the outstation to send an inform to the
        Instation after this time whether or not any reply data has
        changed. If no reply data has changed the inform
        will only contain the timestamp object (utcType2OutstationTime).
        If non-zero then
        1) this value should be greater than or equal to
        the utcType2ScootSampleReportInterval
        2) this time should be greater than the Retry Delay * Retry Count.
        The purpose of this object is to allow the instation to process reply
        data sooner that it might otherwise be able to do so. The instation
        can safely process data in a timely fashion and only delay
        processing of data when the reply data has truly been delayed.
        Default: 0
        "
        ::= { utcType2Config 9 }
-- 1.3.6.1.4.1.13267.3.2.2.10
utcType2ReplyByExceptionResendHoldoff OBJECT-TYPE
    SYNTAX Integer32 (0..180)
    UNITS "Seconds"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "This defines the time to wait before sending data when
        the Retry Count number of attempts have all failed.
        (i.e. when previous notifications have not been acknowledged
        by the instation).
        This is provided to prevent network overloading.
        The holdoff time starts when all the configured retries have timed
        out. A value of 0 causes the re-send to occur straight away.

```

```

        The default value is 1 second."
        ::= { utcType2Config 10 }
-- 1.3.6.1.4.1.13267.3.2.3
utcType2Status OBJECT IDENTIFIER ::= { utmcFullUTCType2 3 }
-- 1.3.6.1.4.1.13267.3.2.3.1
utcType2ScootDetectorCount OBJECT-TYPE
    SYNTAX Integer32 (0..32)
    UNITS "N.O Detectors"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "The number of SCOOT detectors processed by the outstation
        "
        ::= { utcType2Status 1 }
-- 1.3.6.1.4.1.13267.3.2.3.2
utcType2OutstationTime OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (15))
    UNITS "YYYYMMDDHHmmssZ"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object returns the current time and date the outstation is set
        to. It is formatted as YYYYMMDDHHmmssZ.
        Z indicates zulu or GMT
        This format is based upon ExtUTCTime from RFC:2578 and is the
        format used in other SNMP and UTMC timestamps.
        NOTE: The reading of this object will be affected by network delay.
        "
        ::= { utcType2Status 2 }
-- 1.3.6.1.4.1.13267.3.2.4
utcType2Control OBJECT IDENTIFIER ::= { utmcFullUTCType2 4 }
-- 1.3.6.1.4.1.13267.3.2.4.1
utcType2OperationMode OBJECT-TYPE
    SYNTAX INTEGER
        {
            standalone(1),
            monitor(2),
            utccontrol(3)
        }
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Tells the Outstation what mode to operate in;
        tells the Instation the current mode of the Outstation.
        The Outstation will only accept changes to the next
        greater or any lesser value i.e. standalone to monitor
        or monitor to utccontrol the outstation must
        reject (as bad value?) requests that increment the value
        by more than 1.
        In standalone mode all output bits are set to zero. Reply bits
        are not sent but can be polled.
        In monitor mode all output bits are set to zero. The OTU sends
    "

```

inform requests to the Instation as defined elsewhere in this MIB.
 In utccontrol mode inform requests are sent as in monitor mode
 and output bits are controlled by an external system."

REFERENCE

"TR2523A Section 4"

::= { utcType2Control 1 }

-- 1.3.6.1.4.1.13267.3.2.4.2

utcControlTable OBJECT-TYPE

SYNTAX SEQUENCE OF UtcControlEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Control Table"

::= { utcType2Control 2 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1

utcControlEntry OBJECT-TYPE

SYNTAX UtcControlEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry into Control Table"

INDEX { utcControlTimeStamp, utcControlSiteID }

::= { utcControlTable 1 }

UtcControlEntry ::=

SEQUENCE {

utcControlTimeStamp

UTCType2TimeStamp,

utcControlSiteID

UTCType2SiteID,

utcControlDX

UTCType2TruthValue,

utcControlDn

UTCType2BitMask,

utcControlFn

UTCType2BitMask,

utcControlSFn

UTCType2BitMask,

utcControlPV

UTCType2TruthValue,

utcControlPX

UTCType2TruthValue,

utcControlSO

UTCType2TruthValue,

utcControlSG

UTCType2TruthValue,

utcControlLO

UTCType2TruthValue,

utcControlLL

UTCType2TruthValue,

utcControlTS

UTCType2TruthValue,

utcControlFM

```

        UTCType2TruthValue,
    utcControlTO
        UTCType2TruthValue,
    utcControlHI
        UTCType2TruthValue,
    utcControlCP
        UTCType2TruthValue,
    utcControlEP
        UTCType2TruthValue,
    utcControlGO
        UTCType2TruthValue,
    utcControlFF
        UTCType2TruthValue,
    utcControlMO
        UTCType2TruthValue
    }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.1
utcControlTimeStamp OBJECT-TYPE
    SYNTAX UTCType2TimeStamp
    UNITS "Seconds"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Seconds of Day
        Timestamp value N on Control will cause the outstation
        to set the item at N-2 seconds past midnight, i.e. with
        timestamp of 63, the specified item would be activated at
        time 00:01:01.
        A timestamp value of 1 should be taken to mean 'now'.
        "
    ::= { utcControlEntry 1 }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.2
utcControlSiteID OBJECT-TYPE
    SYNTAX UTCType2SiteID
    UNITS "SCN"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Identifies which, of possibly several, equipment at this site the
        object should be applied to. The format is a Free Text ASCII String
        Typically this could be an SCN, IP address, or a number."
    ::= { utcControlEntry 2 }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.3
utcControlDX OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Common Demand Bit"
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Condition 1 on the DX control bit
        shall simulate the operation of
        detector inputs to the controller
    
```

from detector equipment on vehicle actuated stages and, where specified, on pedestrian stages by simulating demands or demands/extensions for selected phases associated with each of the stages. Exceptionally, (where specified), certain stages may be excluded from this common demand. DX shall not inhibit the operation of the pedestrian push buttons and/or vehicle or pedestrian detectors.

REFERENCE

"TR2523A Section 4.4.3"

::= { utcControlEntry 3 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.4

utcControlDn OBJECT-TYPE

SYNTAX UTCType2BitMask

UNITS "Stage Demand Bits"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Individual Computer Stage Demand Bits (D1, D2 etc). Where specified in the Works Specification certain stages may be demand dependent each with its own demand bit.
Logic Conditions for Stage Demand Bits. Condition 1 on a stage demand bit (D1, D2 etc.) shall simulate the operation of a detector by simulating the demands and extensions for selected phase(s) associated with the stage."

REFERENCE

"TR2523A Section 4.4.14"

::= { utcControlEntry 4 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.5

utcControlFn OBJECT-TYPE

SYNTAX UTCType2BitMask

UNITS "Force Bits"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Condition 1 shall force the controller to make an immediate change to the selected stage or shall hold a selected stage subject to the following conditions:

a) if the selected stage does not have rightofway then condition 1 on the force bit for that stage, and no other, shall cause a forced change to that stage provided that a demand exists or is assumed to exist for the stage;

b) if the controller is in an intergreen or a minimum green period, the change to the selected stage shall be deferred until the expiry of the minimum green period, provided that the force condition still exists;

c) if the selected stage has already appeared, condition 1 on the force bit for that stage shall reset the phase maximum timers and hold that stage for so long as the condition 1 is received, provided that gap changes to another demanded stage are prevented by vehicle extensions (e.g. either by control demand signals or from local detectors).

REFERENCE

"TR2523A Section 4.4.18"

::= { utcControlEntry 5 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.6

utcControlSFn OBJECT-TYPE

SYNTAX UTCType2BitMask

UNITS "Switch Facility Bits"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Condition 1 shall switch a specified miscellaneous facility, (e.g. a regulatory traffic sign). Interfacing directly to the specified OTU output terminal or via the controller may provide this facility."

REFERENCE

"TR2523A Section 4.4.24"

::= { utcControlEntry 6 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.7

utcControlPV OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Hold Vehicle Bit"

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Condition 1 shall prevent the
    appearance of the pedestrian stage
    by the imposition of a hold
    condition on the vehicle stage. All
    pedestrian demands which have
    not been served, or which occur
    during the hold period, shall be
    stored and allowed to mature in a
    normal manner when the PV signal
    ceases."
REFERENCE
    "TR2523A Section 4.4.29"
::= { utcControlEntry 7 }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.8
utcControlPX OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Pedestrian Demand Bit"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Condition 1 shall demand the
    pedestrian phase. This facility
    should function even if the output
    from the kerbside detector is
    inactive."
REFERENCE
    "TR2523A Section 4.4.30"
::= { utcControlEntry 8 }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.9
utcControlSO OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Solar Switch Override Bit"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "Condition 1 shall switch the traffic
    signals to the nondimmed
    condition, overriding the Solar
    Switch.
    Condition 0 shall not override the
    solar switch."
REFERENCE
    "TR2523A Section 4.4.31"
::= { utcControlEntry 9 }
-- 1.3.6.1.4.1.13267.3.2.4.2.1.10
utcControlSG OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "CLF Group Timer Sync Signal Bit"
MAX-ACCESS read-write

```

STATUS current

DESCRIPTION

"Receipt of an external signal, having the series message format 1, 0, 1 (received over three consecutive transmission message cycles), shall cause the CLF to commence the relevant plan cycle timing from the start of the first group within 1 second 5% of the 0 to 1 transition of the synchronising message. The Group Timer synchronising signal shall take effect at the receipt of the second 1 providing the Group Timer synchronising signal has been correctly received."

REFERENCE

"TR2523A Section 4.4.33"

::= { utcControlEntry 10 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.11

utcControlLO OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Signal Aspect On Bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Where a condition 1 exists for a minimum of 10 seconds, the signals shall switch on in accordance with the Start Up Sequence.. Where a condition 0 is present for a minimum of 10 seconds, the signals shall switch off during a nominated stage, provided that all minimum running periods have expired."

REFERENCE

"TR2523A Section 4.4.35"

::= { utcControlEntry 11 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.12

utcControlLL OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Local Linking Inhibit Bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Condition 1 shall inhibit local linking between parallel stage streams, or other local links as specified in the Works Specification."

REFERENCE

"TR2523A Section 4.4.36"

::= { utcControlEntry 12 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.13

utcControlTS OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Time Sync Signal Bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Receipt of an external signal, having the series message format 1, 0, 1 (received over three consecutive transmission message cycles), shall cause the controller clock to reset to 00:00 hours or other configured time to the nearest second. The controller synchronising signal shall take effect at the receipt of the second 1."

REFERENCE

"TR2523A Section 4.4.37"

::= { utcControlEntry 13 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.14

utcControlFM OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Fall Back Selection Bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"When the signal controller is not in the UTC mode, condition 1 shall inhibit CLF mode and cause the controller to revert a lower priority method of traffic control, e.g. vehicle actuated or fixed time. Condition 0 shall have no effect."

REFERENCE

"TR2523A Section 4.4.39"

::= { utcControlEntry 14 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.15

utcControlTO OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Take Over Bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This facility shall allow control to be accepted from a remote source. While the TO bit is set to logic 0 (inactive condition) the controller

shall ignore the control bits specified in an associated works specification
 Where an ancillary MOVA unit is specified and control is via the UTC interface, control shall only be operational when the Take Over bit (logic condition 1) is present"

REFERENCE
 "TR2523A Section 4.4.40"
 ::= { utcControlEntry 15 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.16
 utcControlHI OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Hurry Call Inhibit Bit"
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "Logic condition 1 shall inhibit Hurry Call Requests."
 REFERENCE
 "TR2523A Section 4.4.42"
 ::= { utcControlEntry 16 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.17
 utcControlCP OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Close Car Park Bit"
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "Logic condition 1 shall close the car park."
 REFERENCE
 "TR2523A Section 4.4.44"
 ::= { utcControlEntry 17 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.18
 utcControlEP OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Bus Extension Permit bit"
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION
 "Condition '1' shall permit the controller to extend the current stage in order to service a bus priority demand overriding normal forced stage bit control logic."
 ::= { utcControlEntry 18 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.19
 utcControlGO OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Gap Out bit"
 MAX-ACCESS read-write
 STATUS current

DESCRIPTION

"Gap-out bit. Condition '1' shall permit the controller to move from the current stage for which the force bit is active to another stage for which the force bit is active. The conditions under which it is permitted to do this is when there is no demand for the current stage and there is demand for one of the other stages for which the force bit is present. The stage chosen shall be the next eligible stage in cyclic order. Normal stage change restrictions apply."

::= { utcControlEntry 19 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.20

utcControlFF OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Signals Flashing Control bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Where a condition '1' exists for a minimum of 10 seconds, the signals shall be set to flashing amber during a nominated stage provided that all minimum running periods have expired. Where a condition '0' exists for a minimum of 10 seconds, the signals shall switch on in accordance with the start up sequence.

This facility is only specified for export (non-UK) applications."

::= { utcControlEntry 20 }

-- 1.3.6.1.4.1.13267.3.2.4.2.1.21

utcControlMO OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "MOVA Override Control bit"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Condition '1' will enable MOVA to send force bits to the traffic controller in place of the UTC control bits.

Condition '0' will select normal UTC operation i.e. all the UTC control bits will be sent to the traffic controller.

If MOVA is not available then UTC will remain in control.

For Multi Stream MOVA, a site is considered to be the same as a MOVA stream.

"

::= { utcControlEntry 21 }

-- 1.3.6.1.4.1.13267.3.2.5

utcType2Reply OBJECT IDENTIFIER ::= { utmcFullUTCType2 5 }

-- 1.3.6.1.4.1.13267.3.2.5.1

utcReplyTable OBJECT-TYPE

SYNTAX SEQUENCE OF UtcReplyEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Reply Table"

::= { utcType2Reply 1 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1

utcReplyEntry OBJECT-TYPE

SYNTAX UtcReplyEntry

```

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
    "Reply Table Entry"
INDEX { utcReplyTimeStamp, utcReplySiteID }
 ::= { utcReplyTable 1 }
UtcReplyEntry ::=
SEQUENCE {
    utcReplyTimeStamp
        UTCType2TimeStamp,
    utcReplySiteID
        UTCType2SiteID,
    utcReplyGn
        UTCType2BitMask,
    utcReplyGX
        UTCType2TruthValue,
    utcReplyDF
        UTCType2TruthValue,
    utcReplyFC
        UTCType2TruthValue,
    utcReplySCn
        UTCType2BitMask,
    utcReplyHC
        UTCType2TruthValue,
    utcReplyWI
        UTCType2TruthValue,
    utcReplyPC
        UTCType2TruthValue,
    utcReplyPR
        UTCType2TruthValue,
    utcReplyCG
        UTCType2TruthValue,
    utcReplyGR1
        UTCType2TruthValue,
    utcReplySDn
        UTCType2BitMask,
    utcReplyMC
        UTCType2TruthValue,
    utcReplyCF
        UTCType2TruthValue,
    utcReplyLE
        UTCType2TruthValue,
    utcReplyRR
        UTCType2TruthValue,
    utcReplyLFn
        UTCType2BitMask,
    utcReplyRF1
        UTCType2TruthValue,
    utcReplyRF2
        UTCType2TruthValue,
    utcReplyEV
        UTCType2TruthValue,

```

```

utcReplyVC
    UTCType2TruthValue,
utcReplyVO
    UTCType2TruthValue,
utcReplyGPn
    UTCType2BitMask,
utcReplyVQ
    UTCType2TruthValue,
utcReplyCA
    UTCType2TruthValue,
utcReplyCR
    UTCType2TruthValue,
utcReplyCL
    UTCType2TruthValue,
utcReplyCSn
    UTCType2BitMask,
utcReplyTF
    UTCType2TruthValue,
utcReplyVSn
    OCTET STRING,
utcReplyCO
    UTCType2TruthValue,
utcReplyEC
    UTCType2TruthValue,
utcReplyCS
    UTCType2TruthValue,
utcReplyFR
    UTCType2TruthValue,
utcReplyBDn
    UTCType2BitMask,
utcReplyTPn
    UTCType2BitMask,
utcReplySB
    UTCType2TruthValue,
utcReplyLC
    UTCType2TruthValue,
utcReplyMR
    UTCType2TruthValue,
utcReplyMF
    UTCType2TruthValue,
utcReplyML
    UTCType2TruthValue
    }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.1
utcReplyTimeStamp OBJECT-TYPE
    SYNTAX UTCType2TimeStamp
    UNITS "Seconds of Day"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Seconds Of Day
        Timestamp value N on reply will indicate the changes have

```

```

        been detected during the one second period ending N-2 seconds
        past midnight.
        A timestamp value of 1 should be taken to mean 'now'. "
        ::= { utcReplyEntry 1 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.2
utcReplySiteID OBJECT-TYPE
    SYNTAX UTCType2SiteID
    UNITS "SCN"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Identifies which, of possibly several, equipment at this site the
        object should be applied to. The format is a Free Text ASCII String
        Typically this could be an SCN, IP address, or a number."
        ::= { utcReplyEntry 2 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.3
utcReplyGn OBJECT-TYPE
    SYNTAX UTCType2BitMask
    UNITS "Stage Confirmation Bits"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition 1 confirms that a
        particular stage, or phase if
        specified is running.
        G1 and G2 shall normally be
        returned simultaneously to indicate
        that one of the following has
        occurred:
        a) the mains supply to the signal
        aspects is off;
        b) manual method of traffic
        control is either in operation or
        requested;
        c) The traffic controller is
        switched off;
        d) The traffic controller has failed
        or shut down due to a fault;
        e) The interface between the
        OTU and the controller has
        been disconnected.
        "
    REFERENCE
        "TR2523A Section 4.5.5"
        ::= { utcReplyEntry 3 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.4
utcReplyGX OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Vehicle State Green Confirmation Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION

```

```

"Condition 1 confirms that a green
signal is displayed to vehicles on a
standalone controller. When the
signals are not on stage green, or
when the controller or signals are
switched off, the indication returned
shall be condition 0."
REFERENCE
"TR2523A Section 4.5.7"
::= { utcReplyEntry 4 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.5
utcReplyDF OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Detector Fault Monitor Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Condition 1 confirms that the
detector fault monitor system
indicates a detector failure."
REFERENCE
"TR2523A Section 4.5.8"
::= { utcReplyEntry 5 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.6
utcReplyFC OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "FallBack Selection Confirmation Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Condition 1 confirms that the Fall
Back selection facility has been
introduced."
REFERENCE
"TR2523A Section 4.5.9"
::= { utcReplyEntry 6 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.7
utcReplySCn OBJECT-TYPE
SYNTAX UTCType2BitMask
UNITS "Switch Facility Confirmation Bits"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"Condition 1 confirms that a
particular Switch Facility has been
introduced."
REFERENCE
"TR2523A Section 4.5.10"
::= { utcReplyEntry 7 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.8
utcReplyHC OBJECT-TYPE
SYNTAX UTCType2TruthValue

```

```

UNITS "Hurry Call Confirmation Or Request Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that a Hurry
    Call request has been requested or
    is being actioned, as specified in an
    associated Works Specification."
REFERENCE
    "TR2523A Section 4.5.11"
::= { utcReplyEntry 8 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.9
utcReplyWI OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Wait Indicator Confirm Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that the
    WAIT indicator (Pelican) or the
    Pedestrian Demand Accepted
    (Puffin) is energised at stand-alone
    crossings."
REFERENCE
    "TR2523A Section 4.5.12"
::= { utcReplyEntry 9 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.10
utcReplyPC OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Pedestrian Stage Green Confirm Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that the
    pedestrian green signal is
    energised. Condition 0 shall be
    given when the controller or signals
    are switched off. This can apply to
    junction or standalone facilities."
REFERENCE
    "TR2523A Section 4.5.13"
::= { utcReplyEntry 10 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.11
utcReplyPR OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Puffin Pedestrian Clearance Period Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that the
    pedestrian clearance period is
    operative. Condition 0 shall be

```

given when the controller or signals are switched off."

REFERENCE

"TR2523A Section 4.5.14"

::= { utcReplyEntry 11 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1.12

utcReplyCG OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "CLF Group Timer Sync Confirm Bit"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A signal shall be returned to the OTU/controller interface when the synchronising signal has been correctly received and actioned. This reply signal (condition 1) shall be normally maintained for a period of 3 seconds 1 second or as specified in the Works Specification.

As an option the CG bit may confirm the time of day and day of week in the controller clock. The CG bit may be set to condition 1 (active) at a predetermined period after the controller synchronisation time. The length of time the signal is held active shall indicate the day of the week as follows:

- Sunday 3 seconds
- Monday 5 seconds
- Tuesday 7 seconds
- Wednesday 9 seconds
- Thursday 11 seconds
- Friday 13 seconds
- Saturday 15 seconds

"

REFERENCE

"TR2523A Section 4.5.15"

::= { utcReplyEntry 12 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1.13

utcReplyGR1 OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Group 1 Indication Bit"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"That CLF is in the first group. This reply signal (condition 1) shall be maintained for a period of three seconds 1 second."

```

REFERENCE
    "TR2523A Section 4.5.17"
    ::= { utcReplyEntry 13 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.14
utcReplySDn OBJECT-TYPE
    SYNTAX UTCType2BitMask
    UNITS "Stage Demands Bits"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition 1 confirms that a
        demand exists for a stage."
REFERENCE
    "TR2523A Section 4.5.18"
    ::= { utcReplyEntry 14 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.15
utcReplyMC OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Manual Control Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition 1 confirms that Manual
        Control is either in operation or
        requested as specified in an
        associated Works Specification."
REFERENCE
    "TR2523A Section 4.5.19"
    ::= { utcReplyEntry 15 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.16
utcReplyCF OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Controller Fault Indication Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition 1 confirms that an entry
        is in the system fault log."
REFERENCE
    "TR2523A Section 4.5.20"
    ::= { utcReplyEntry 16 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.17
utcReplyLE OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Signal Aspects Extinguished Indication Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "That the mains supply to the signal
        aspects has been interrupted by:
        a) operation of the signal aspect
        switch, or;
    
```

b) the signal aspect fuse being blown, or;
the controller mains supply being off (only in the case of a separately powered OTU).
This may include part time signal operation."

REFERENCE

"TR2523A Section 4.5.21"

::= { utcReplyEntry 17 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1.18

utcReplyRR OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Remote Reconnect Bit"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"As an optional facility, the controller may be released from remote control due to manual intervention and should be specified in an associated Works Specification. Condition 1 shall request release and condition 0 shall be returned to request reestablishment of remote control (see Clause 4.5.19)."

REFERENCE

"TR2523A Section 4.5.24"

::= { utcReplyEntry 18 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1.19

utcReplyLFn OBJECT-TYPE

SYNTAX UTCType2BitMask

UNITS "Signal Aspect Failure Bit"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Condition 1 confirms that one or more traffic signal aspects have failed, where these are monitored."

REFERENCE

"TR2523A Section 4.5.25"

::= { utcReplyEntry 19 }

-- 1.3.6.1.4.1.13267.3.2.5.1.1.20

utcReplyRF1 OBJECT-TYPE

SYNTAX UTCType2TruthValue

UNITS "Vehicle Red Signal Aspect Failure 1 Bit"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Condition 1 confirms that at least one vehicle red signal aspect has been accepted as failed where

these are monitored for Part Time or Pedestrian Audible/Tactile Control."

REFERENCE
 "TR2523A Section 4.5.26"
 ::= { utcReplyEntry 20 }
 -- 1.3.6.1.4.1.13267.3.2.5.1.1.21
 utcReplyRF2 OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Vehicle Red Signal Aspect Failure 2 Bit"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Condition 1 confirms that a second vehicle red signal aspect has been accepted as failed on an approach, or a vehicle red signal aspect feed has failed where these are monitored for Part Time or Pedestrian/Audible Tactile Control or the Red Signal aspect monitor has failed."
 REFERENCE
 "TR2523A Section 4.5.27"
 ::= { utcReplyEntry 21 }
 -- 1.3.6.1.4.1.13267.3.2.5.1.1.22
 utcReplyEV OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Emergency Vehicle Bit"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "Condition 1 confirms that the controller is servicing a priority call, other than a hurry call."
 REFERENCE
 "TR2523A Section 4.5.28"
 ::= { utcReplyEntry 22 }
 -- 1.3.6.1.4.1.13267.3.2.5.1.1.23
 utcReplyVC OBJECT-TYPE
 SYNTAX UTCType2TruthValue
 UNITS "Vehicle Count Bit"
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "A count of the number of vehicle pulses scaled by a predetermined scale factor. Each time the reply bit changes state, this indicates that the predetermined number of vehicles have been counted at the outstation. The predetermined scale factor is defined separately for the outstation and is outside the scope of this MIB."
 REFERENCE

```

"TR2523A Section 4.5.29"
 ::= { utcReplyEntry 23 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.24
utcReplyVO OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Vehicle Occupancy Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "An occupancy detector samples a vehicle loop 25
        times per second. Every time the loop is occupied
        a counter is incremented. Bit 5 of the counter
        (starting at bit 0) is returned to the UTC
        instation. This means that the bit changes state
        every time the unit has seen thirty-two
        twenty-fifths of a second of occupancy."
 ::= { utcReplyEntry 24 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.25
utcReplyGPn OBJECT-TYPE
    SYNTAX UTCType2BitMask
    UNITS "General Purpose Reply bits"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "This object is used to return miscellaneous
        equipment reply bit states to the instation.
        The purpose of this object is to allow the
        protocol to handle reply bit functions that
        are not represented by other MIB objects.
        In essence, it implements for reply functions
        what the switch facility object implements
        for control.
        "
 ::= { utcReplyEntry 25 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.26
utcReplyVQ OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Queue Detector Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition 1 confirms that the
        Vehicle Queue Detector indicates a
        queue state."
    REFERENCE
        "TR2523A Section 4.5.30"
 ::= { utcReplyEntry 26 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.27
utcReplyCA OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Car Park Occupancy Threshold Exceeded Bit"
    MAX-ACCESS read-only

```

```

STATUS current
DESCRIPTION
    "Condition 1 confirms that the car
    park occupancy threshold is
    exceeded."
REFERENCE
    "TR2523A Section 4.5.31"
::= { utcReplyEntry 27 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.28
utcReplyCR OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Queue At Car Park Reservoir Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that a queue
    state exists at the car park entry
    reservoir."
REFERENCE
    "TR2523A Section 4.5.32"
::= { utcReplyEntry 28 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.29
utcReplyCL OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Car Park Closed Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 confirms that the car
    park is closed."
REFERENCE
    "TR2523A Section 4.5.33"
::= { utcReplyEntry 29 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.30
utcReplyCSn OBJECT-TYPE
SYNTAX UTCType2BitMask
UNITS "Car Park Information Bit"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition 1 indicates the state of
    specified signs associated with the
    car park."
REFERENCE
    "TR2523A Section 4.5.34"
::= { utcReplyEntry 30 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.31
utcReplyTF OBJECT-TYPE
SYNTAX UTCType2TruthValue
UNITS "Handset Connected Bit"
MAX-ACCESS read-only
STATUS current

```

```

DESCRIPTION
    "Condition 1 confirms that the
    handset equipment is connected to
    the Terminal interface."
REFERENCE
    "TR2523A Section 4.5.35"
::= { utcReplyEntry 31 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.32
utcReplyVSn OBJECT-TYPE
    SYNTAX OCTET STRING (SIZE (1..100))
    UNITS "SCOOT Detector Output Presence Bit"
    MAX-ACCESS read-only
    STATUS current
DESCRIPTION
    "Condition 1 is the active output
    state on a SCOOT detector.
    Note: These are four sample
    bits per detector per second.
    The encoding is:
      1   2   3
    (D2 D1) (D4 D3) (0 D5) - SCOOT samples for second N
      4   5   6
    (D2 D1) (D4 D3) (0 D5) - SCOOT samples for second N-1
      7   8   9
    (D2 D1) (D4 D3) (0 D5) - SCOOT samples for second N-2
      z
    (xx yy) represents an octet of data,
    where xx is the most significant nibble and
    yy is the least significant nibble
    z is the octet number within the octet string
    D1 represents the 4 samples for SCOOT detector 1
    taken over a second, with the most recent
    sample in the least significant bit of the nibble.
    D2 represents the 4 samples for SCOOT detector 2, etc
    0 is a packing nibble, only required where there are
    an odd number of SCOOT detectors.
    The timestamp on the reply table entry will represent
    the time at which the accumulation of the samples is concluded.
    The maximum number of seconds of data in an octet string
    is configured by utcType2ScootSampleReportInterval, although
    a report may contain less than this, where the data is
    sent before the full number of seconds have been accumulated."
REFERENCE
    "TR2523A Section 4.5.36"
::= { utcReplyEntry 32 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.33
utcReplyCO OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Cabinet Door Open Bit"
    MAX-ACCESS read-only
    STATUS current
DESCRIPTION

```

```

        "Condition 1 confirms that the
        cabinet door is open."
    REFERENCE
        "TR2523A Section 4.5.37"
    ::= { utcReplyEntry 33 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.34
utcReplyEC OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Bus Extension Confirm bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' confirms that the controller is extending the current
        stage to service a bus priority demand."
    ::= { utcReplyEntry 34 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.35
utcReplyCS OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Time Synchronisation Confirm bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "A signal shall be returned to the OTU/controller interface when the
        TS synchronising signal has been correctly received and actioned.
        This reply signal (condition '1') shall be normally maintained for a
        period of 3 seconds 1 second or as specified in the Works
        Specification. As an option the CS bit may confirm the time of day
        and day of week in the controller clock. The CS bit may be set to
        condition '1' (active) at a predetermined period after the controller
        synchronisation time.
        The length of time the signal is held active shall indicate the day of
        the week as follows:
        Sunday 3 seconds
        Monday 5 seconds
        Tuesday 7 seconds
        Wednesday 9 seconds
        Thursday 11 seconds
        Friday 13 seconds
        Saturday 15 seconds
        Or as specified in the works specification.
        "
    ::= { utcReplyEntry 35 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.36
utcReplyFR OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Signals Flashing Confirm bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' confirms that the controller signals are in flashing
        amber mode.
        This bit is only specified for export (non-UK) applications."

```

```

        ::= { utcReplyEntry 36 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.37
utcReplyBDn OBJECT-TYPE
    SYNTAX UTCType2BitMask
    UNITS "Bus Detected bits"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' for a single bit in the bit mask confirms that a bus
        has been detected"
        ::= { utcReplyEntry 37 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.38
utcReplyTPn OBJECT-TYPE
    SYNTAX UTCType2BitMask
    UNITS "RTIG Traffic Light Priority Trigger Point bits"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' confirms activation of an RTIG Traffic Light Priority
        trigger point.
        TP1 = Registration
        TP2 = Request
        TP3 = Clear
        See 'Specification for the Radio Link Protocol and Transmission
        Methodology for RTIG Traffic Light Priority and Display Clear
        Down' for more information.
        "
        ::= { utcReplyEntry 38 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.39
utcReplySB OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Solar Brightness bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' confirms traffic signals are non-dimmed."
        ::= { utcReplyEntry 39 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.40
utcReplyLC OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "Local Link Inhibit Confirm bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' confirms inhibition of local linking between
        parallel stage streams, or other local links as specified in the
        Works Specification."
        ::= { utcReplyEntry 40 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.41
utcReplyMR OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "MOVA Override Reply Bit"

```

```

MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Condition '1' confirms that the MO bit is being actioned by the
    outstation. A condition '1' can also mean that MOVA has been told
    to override UTC control by a local source.
    NB: There may be local conditions which mean that although MO
    is active and MR has been returned, MOVA is not actually
    controlling the controller e.g. MOVA many not have been
    configured, MOVA is in its warm-up cycles or MOVA is off-line.
    The ML bit will confirm that MOVA is in the on or off-line condition.
    "
    ::= { utcReplyEntry 41 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.42
utcReplyMF OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "MOVA Fault Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' means that MOVA has detected a fault, this is
        equivalent to the MOVA Phone-Home condition.
        The MOVA unit will report this fault when it has detected an
        Unrecoverable error.
        "
        ::= { utcReplyEntry 42 }
-- 1.3.6.1.4.1.13267.3.2.5.1.1.43
utcReplyML OBJECT-TYPE
    SYNTAX UTCType2TruthValue
    UNITS "MOVA Online Bit"
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Condition '1' means that MOVA is on-line and controlling the
        traffic controller.
        If MOVA is not available, this value is set to '0'.
        "
        ::= { utcReplyEntry 43 }
-- 1.3.6.1.4.1.13267.3.2.6
utcType2Notifications OBJECT IDENTIFIER ::= { utmcFullUTCType2 6 }
-- 1.3.6.1.4.1.13267.3.2.6.1
utcReplyByException NOTIFICATION-TYPE
    STATUS current
    DESCRIPTION
        "This is the Notification used to send reply by exception data to
        the instation.
        "
        ::= { utcType2Notifications 1 }
-- 1.3.6.1.4.1.13267.3.2.7
utcType2Conformance OBJECT IDENTIFIER ::= { utmcFullUTCType2 7 }
-- 1.3.6.1.4.1.13267.3.2.7.1
utcType2Groups OBJECT IDENTIFIER ::= { utcType2Conformance 1 }

```

```

-- 1.3.6.1.4.1.13267.3.2.7.1.1
utcNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS { utcReplyByException }
    STATUS current
    DESCRIPTION
        "This group holds all the objects in the notification branch."
    ::= { utcType2Groups 1 }
-- 1.3.6.1.4.1.13267.3.2.7.1.2
utcVersionGroup OBJECT-GROUP
    OBJECTS { utcType2AppVersion, utcType2HardwareID,
        utcType2HardwareType, utcType2VendorID, utcType2AppPartNumber,
        utcType2MIBVersion }
    STATUS current
    DESCRIPTION
        "This group holds all the objects in the version branch"
    ::= { utcType2Groups 2 }
-- 1.3.6.1.4.1.13267.3.2.7.1.3
utcConfigGroup OBJECT-GROUP
    OBJECTS { utcType2ConfigLastChanged, utcType2InstationAddress,
        utcType2InstationPort, utcType2OperationModeTimeout,
        utcType2ReplyByExceptionRetryDelay,
        utcType2ReplyByExceptionRetryCount,
        utcType2ScootSampleReportInterval,
        utcType2ReplyByException,
        utcType2ReplyByExceptionKeepAlive,
        utcType2ReplyByExceptionResendHoldoff    }
    STATUS current
    DESCRIPTION
        "Group of objects not in reply or control."
    ::= { utcType2Groups 3 }
-- 1.3.6.1.4.1.13267.3.2.7.1.4
utcStatusGroup OBJECT-GROUP
    OBJECTS { utcType2ScootDetectorCount, utcType2OutstationTime }
    STATUS current
    DESCRIPTION
        "Description."
    ::= { utcType2Groups 4 }
-- 1.3.6.1.4.1.13267.3.2.7.1.5
utcControlGroup OBJECT-GROUP
    OBJECTS { utcControlSiteID, utcControlIDX, utcControlDn, utcControlFn,
        utcControlSFn, utcControlIPV, utcControlIPX, utcControlISO, utcControlISG,
        utcControlLO, utcControlLL, utcControlTS, utcControlFM, utcControlTO,
        utcControlHI, utcControlTimeStamp, utcControlIEP, utcControlGO,
        utcControlFF, utcControlMO, utcControlCP, utcType2OperationMode }
    STATUS current
    DESCRIPTION
        "This group holds all the Control Objects"
    ::= { utcType2Groups 5 }
-- 1.3.6.1.4.1.13267.3.2.7.1.6
utcReplyGroup OBJECT-GROUP
    OBJECTS { utcReplySiteID, utcReplyGn, utcReplyGX, utcReplyDF,
        utcReplyFC, utcReplySCn, utcReplyHC, utcReplyWI, utcReplyPC,

```

```

    utcReplyPR, utcReplyCG, utcReplyGR1, utcReplySDn, utcReplyMC,
    utcReplyCF, utcReplyLE, utcReplyRR, utcReplyLFn, utcReplyRF1,
    utcReplyRF2, utcReplyEV, utcReplyVC, utcReplyVQ, utcReplyCA,
    utcReplyCR, utcReplyCL, utcReplyCSn, utcReplyTF, utcReplyVSn,
    utcReplyVO, utcReplyEC, utcReplyCS, utcReplyFR, utcReplyBDn,
    utcReplyTPn, utcReplySB, utcReplyLC, utcReplyMR, utcReplyMF,
    utcReplyML, utcReplyCO, utcReplyTimeStamp, utcReplyGPn }
    STATUS current
    DESCRIPTION
        "This group holds all the Reply Objects"
        ::= { utcType2Groups 6 }
-- 1.3.6.1.4.1.13267.3.2.7.2
utcType2Compliance OBJECT IDENTIFIER ::= { utcType2Conformance 2 }
-- 1.3.6.1.4.1.13267.3.2.7.2.1
utcType2MandatoryUG405Compliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "This is the Compliance module for mandatory Objects that must
        Be supported, "
    MODULE utmcFullUTCType2
        MANDATORY-GROUPS { utcVersionGroup, utcConfigGroup,
        utcControlGroup, utcStatusGroup, utcReplyGroup,
        utcNotificationGroup }
        ::= { utcType2Compliance 1 }

    END
--
-- UTMC-UTMCFULLUTCTYPE2-MIB-1rc2.my
--

```

F Definitions of registered IDL scripts (Normative)

F.1 Introduction

F.1.1 IDL scripts for use in UTM systems are presented in text form below.

F.1.2 These standardised scripts should be used wherever possible to manage the data passed across a CORBA link within a UTM system.

F.2 UI/001, BCD.idl

```
/*
/**
** This module defines generic binary (octet sequence) types, and arbitrary
** precision floating-point decimal data types.
** For generic coding, the decimal types are a useful alternative to the IDL
** fixed template types which were introduced by CORBA 2.1 and
** GIOP 1.1, since the use of template types introduces a distinct type for
** each combination of precision and scale.
**/
module BCD
{
  /**
  ** A generic octet sequence type.
  **/
  typedef sequence < octet > Binary;
  /**
  ** A floating-point decimal type.
  ** The Binary representation of floating-point values is
  ** designed to support easy conversion to and from strings, while at the
  ** same time minimising the size of the binary representation for GIOP.
  ** A number is converted to the binary representation using the
  ** following algorithm.
  **
  ** Convert the number to a decimal string, with a leading '-' if
  ** negative, and no leading sign if zero or positive.
  ** The string may contain a decimal point. If the number's scale is
  ** greater than zero, the decimal point is followed by the number's decimal
  ** fraction. The number of digits in the fraction indicates the scale;
  ** trailing zeroes should not be truncated as that would reduce the implied
  ** scale.
  ** Convert the string's characters to an octet sequence where each
  ** half-octet represents padding, a decimal digit, negative sign, or
  ** decimal point, as follows:
  **
  ** If the string length is odd, the initial half-octet value is 12.
  ** This is used as padding to ensure an even number of half-octets.
  ** '0' to '9' map to half-octet values 0 to 9.
  ** '.' maps to half-octet value 10.
  ** '-' maps to half-octet value 13.
  **
  ** In each octet of the sequence, the most significant half-octet is
  ** filled first. Thus the value "-9" would be encoded as octet value
  ** 13 * 16 + 9 = 217 .
  **
  **/
}
```

```

** Any program receiving a binary value which is not formed according
** to the above specification should raise a CORBA::DATA_CONVERSION system
** exception.
** Note: a struct is used because it allows overloaded methods to be
** defined in C++ and Java for decimal manipulation. It also guarantees
** that a distinct type is generated for the Java IDL language mapping.
**/
struct Decimal
{
    Binary value;
};
/**
** A floating-point decimal (money) type. Uses the same representation
** as Decimal but may be used where a semantic distinction is
** required.
** Note: a struct is used because it allows overloaded methods to be
** defined in C++ and Java for money manipulation. It also guarantees
** that a distinct type is generated for the Java IDL language mapping.
**/
struct Money
{
    Binary value;
};
};

```

F.3 UI/002, MJD.idl

```

/*
/*
* This module defines date/time types based on the Modified Astronomical
* Julian Date (M.J.D.) standard. The Modified Julian Date is given in
* decimal form, not in hours and minutes.
*
* Source: Norton's Star Atlas and Reference Handbook, 17th edition.
*/
module MJD
{
    /*
    * The Date type encodes a date/time as J.D. - 2,400,000.5,
    * where J.D. = 0 represents Jan 1, 4713 B.C. at Greenwich noon.
    * For example, Greenwich midnight Jan 1, 1970 is M.J.D. 40222.0.
    * This type provides millisecond resolution for dates at least
    * 10,000 years either side of the base value (MJD = 0.0).
    * Note: a struct is used because it allows overloaded methods to be
    * defined in C++ and Java for date manipulation.
    */
    /* The Modified Julian Day or MJD is defined as:
    * (MJDay = JDay - 2400000.5)
    * so that MJD = 0.0 corresponds to midnight between the 16 & 17 November 1858 AD gregorian.
    *
    * The Julian Day Number, Julian Day, or JD of a particular instant of time is the number
    * of days and fractions of a day since 12 hours Universal Time (Greenwich mean noon) on
    * January 1 of the year 4713 BC.
    */
}

```

```

*
* To calculate the MJD use the value from the C runtime library which gives the days from
* the year 1900 and then add the offset back to 1858.
*/
struct Date
{
    double dateValue;
};
/*
* The Time type encodes a time with nanosecond
* resolution. The encoded value timeValue is equivalent
* to Date's dateValue, but only the fractional part of
* timeValue is understood to have significance.
* The integral part should be zero, since any significant digits in
* the integral part reduce the precision of the time value.
* Note: a struct is used because it allows overloaded methods to be
* defined in C++ and Java for time manipulation.
*/
struct Time
{
    double timeValue;
};
/*
* The Timestamp type encodes a date/time with nanosecond
* resolution. The encoded value dateValue + timeValue is equivalent
* to Date's dateValue, but allows for nanosecond resolution.
* To ensure no loss of significance, dateValue must be
* integral (may be negative) and 0 <= timeValue < 1 .
* This type provides nanosecond resolution for dates at least
* 100,000,000,000 years either side of the base value (MJD = 0.0).
*/
struct Timestamp
{
    double dateValue;
    double timeValue;
};
};

```

F.4 UI/003, B-Query.idl

```

/*
***
This file defines the interface to the query mechanism
for UTMC CORBA clients.
*/
#include "TabularResults.idl"
module Query
{
    /**
    The CorbaSQLException structure mimics the
    java.sql.SQLException class contents.
    It will not actually be raised as an exception,
    but be included as a member of the

```

```

CorbaSQLExceptions exception
*/
struct CorbaSQLException
{
    string detailMessage;
    string SQLState;
    long errorCode;
};
/**
The CorbaSQLWarning structure mimics the
java.sql.SQLWarning class contents.
It will not actually be raised as an exception,
but be included as a member of the
CorbaSQLWarnings output parameter of
the method calls defined below.
*/
struct CorbaSQLWarning
{
    string detailMessage;
    string SQLState;
    long errorCode;
};
/**
The CorbaSQLExceptions exception comprises
a sequence of CorbaSQLException structures.
The exception will be raised if one or more
java.sql.SQLException's are thrown within
the method calls.
*/
exception CorbaSQLExceptions
{
    sequence < CorbaSQLException > exceptions;
};
/**
The ClientIdentificationException exception
will be raised if problems are encountered while trying
to identify the incoming client connection.
*/
exception ClientIdentificationException
{
};

typedef sequence < CorbaSQLWarning > CorbaSQLWarnings;
typedef sequence < long > UpdateCounts;
interface SQLStatement
{
    /**
Executes the statement provided as the input
param statement.
A 'result' is either a ResultSet for a query
statement or an update count for non-query
statements. Normally a single result will be
obtained, however if the statement is a complex

```

SQL script, then it may return multiple 'results'
 The resultSets param contains the 'results'
 that returned ResultSets(i.e. rows of data).
 The updateCounts param contains the update
 count for each 'result' that returned an update count
 (usually an UPDATE statement or DDL statement).
 The warnings param contains any
 SQLWarnings that were generated while executing the
 statement.

```
*/
void execute
(
    in string statement,
    out TabularResults::ResultSets resultSets,
    out UpdateCounts updateCount,
    out CorbaSQLWarnings warnings
)
raises (CorbaSQLExceptions, ClientIdentificationException);
/**
```

Executes an insert statement for a Data Object Component and returns the
 identifier for the inserted object.

This will provide a means of inserting anonymous objects e.g.
 commands, log entries, faults , vms messages etc.
 and allow the database to determine the unique identifier for
 the inserted object.

The updateCounts param contains the update
 count for each 'result' that returned an update count
 (usually an UPDATE statement or DDL statement).
 The warnings param contains any
 SQLWarnings that were generated while executing the
 statement.

```
*/
string insertDataObject
(
    in string statement,
    out UpdateCounts updateCount,
    out CorbaSQLWarnings warnings
)
raises (CorbaSQLExceptions, ClientIdentificationException);
};
};
```

F.5 UI/004, B-SessionManagement.idl

```
/*
/*
* This module defines interfaces for session-based creation
* of transient server objects.
*/
module SessionManagement
{
    /*
    * The Session interface represents an authenticated
```

```

* session with a server that implements transient objects
*/
interface Session
{
  /*
  * Create a transient object. The caller must narrow the
  * returned object reference to a type which has been generated by
  * the IDL compiler from the object's IDL interface.
  * The repositoryID represents the repository ID of the
  * CORBA object e.g. "IDL:Subscriptions/SubscriptionRequest:1.0"
  Object createObject(in string repositoryID);
  */
  /*
  * Call made by a client so the server can monitor and timeout its connection
  * The parameter sessionTimeout shall be specified in seconds and
  * shall be within a "default timeout period" range established by the Server
  * Returned value by Server is false if the parameter value is outside range else true.
  */
  boolean registerHeartbeat( in long sessionTimeout );
  /*
  * Called by the client to inform the Server that the Client is still alive.
  */
  void clientHeartbeat();
};
/*
* The Manager interface represents an authentication
* service. It supports explicit session-based authentication for
* servers implementing transient objects. Transient objects may be
* created in the context of an authenticated session.
*/
interface Manager
{
  /*
  * Create an authenticated session. The lifetime of the session is
  * determined by the server based on available resources. It can be
  * assumed that the server will only terminate an active
  * session in exceptional circumstances.
  * Note: the session lifetime is not limited to the duration of the
  * connection (e.g. TCP socket) which was used to create the session.
  */
  Session createSession(in string user, in string password);
  /*
  * frees server resources by ending the session created above
  * The Session parameter passed in identifies the session that will be ended and
  * is the value returned by CreateSession.
  */
  void endSession (in Session sess );
  /*
  * frees server resources by ending the session created above
  * This optional method is used by supervisory applications to forcibly terminate an existing
  * session identified by Name. Name parameter is that provided during the CreateSession.
  */
  void endSessionWithUser( in string user );
};
};

```

F.6 UI/005, B-Subscriptions.idl

```

/*
#include "TabularResults.idl"
/**
This module provides the UTMC10 subscription facility.
It is used both to describe the data that it is desired
to subscribe to and also to provide the interface to
the delivery of the subscribed data via notification
requests.
*/
module Subscriptions
{
    enum DatabaseAction
    {
        INSERT_ACTION,
        DELETE_ACTION,
        UPDATE_ACTION
    };
    /**
    This structure is the unit of notification that will
    be sent as a result of a push event. The fields have
    the following meaning.

    userName. The users login name. This will
    always be the name of the connected client (i.e. Vasp name)
    componentName. The name of the table that
    changed to cause the trigger for the push event.
    viewName. The name of the database view
    that is being used to determine the content and criteria
    for the pushed data.
    action. The code that identifies what type
    of activity is being performed with respect to the view.
    results. The actual data that is being
    pushed.
    */
    struct DataNotification
    {
        string userName;
        string componentName;
        string viewName;
        DatabaseAction action;
        TabularResults::ResultSet results;
    };
    /* sequence of DataNotifications called by pullSequence
    to allow multiple returns on data change
    */
    typedef sequence<DataNotification> DataNotifications;

    /**
    The interface for creating and deleting subscriptions.
    */
    interface SubscriptionRequest
    {

```

```
/**
This method should be called to request a subscription
to data. This requests that any data changes that
meet the view criteria should be pushed.
```

The `componentName` parameter identifies the table which is the 'trigger' to be used for the push.

The `viewName` parameter identifies the view which is to be used for the criteria and content of the pushed data.

The return value will be false if the subscription was successfully created.

```
*/
boolean createSubscription
(
    in string componentName,
    in string viewName
);
```

```
/**
This method should be called to delete a subscription
to data.
```

The `componentName` and `viewName` parameters identify the subscription to delete.

The return value will be false if the subscription was successfully deleted.

```
*/
boolean deleteSubscription
(
    in string componentName,
    in string viewName
);
};
```

```
/**
The NotificationException exception is
used to indicate the unsuccessful completion of
certain operations involved with notification delivery.
The reason field gives details of the
cause of the exception.
```

```
*/
exception NotificationException
{
    string reason;
};
```

```
/**
The NotificationRequest interface is used to
implement the delivery of pushed data using the 'blocked
pull' method of delivery. When an object reference is
created for an interface of this type, UTMC10 will create
a new instance of this object for each client request. This
instance will be for the exclusive use of the creator of
the instance.
```

```

*/
interface NotificationRequest
{
    /**
    This method should be called whenever it is desired
    to be informed that data has been changed.
    The method call will suspend the calling thread of
    execution until such time as there is actually data
    available for the caller. At this point, the thread
    will be resumed, and the method will return the
    DataNotification object that represents
    the pushed data.
    It is intended that this method be called in a
    repeating manner, with the normal state being blocked,
    and when the method returns the client program will
    interpret the result, and immediatly call the method
    again thus suspending the thread once more.
    */
    DataNotification pull
    (
    )raises(NotificationException);
    /**
    * This method is a more efficient version of the standard pull method
    * It returns all available data from multiple inserts/updates in one call
    * rather than the client making a call per updated data
    */
    DataNotifications pullSequence
    (
    )raises(NotificationException);
    /**
    This method should be called when this particular
    NotificationRequest object instance
    is no longer needed. There is no functional
    effect to this method call, but it does free up
    server resources in a more timely manner.
    */
    void releaseObj
    (
    );
};
/**
The NotificationTarget interface defines
the object that a client program will need to implement
to receive notification of changes to data.
*/
interface NotificationTarget
{
    /**
    The inform method is used to provide
    the DataNotification object to the
    client when the data changes.
    */
    void notificationReceived
    (

```

```

        in DataNotification notification
    );
};
/**
The CallbackNotificationRequest
interface is used to provide UTMC10 with the details
of which client object should be used to inform the
client of changed data.
*/
interface CallbackNotificationRequest
{
    /**
    Register a NotificationTarget for callbacks. A
    Notification Target Id number is returned, which
    must be used when deregistering that same
    NotificationTarget.
    A NotificationException is raised if the
    operation did not complete successfully.
    */
    long register
    (
        in NotificationTarget target
    )
    raises (NotificationException);
    /**
    Deregister a NotificationTarget using the
    Notification Target Id that was returned when
    the NotificationTarget was first registered.
    A NotificationException is raised if the
    operation did not complete successfully.
    */
    void deregister
    (
        in long targetId
    )
    raises (NotificationException);
};
};

```

F.7 UI/006, B-TabularResults.idl

```

/*
#include "BCD.idl"
#include "MJD.idl"
*/
* This module defines CORBA IDL constants and types for dynamic result
* set handling, with the intention of supporting easy conversion from
* and to java.sql.ResultSet (or jdbc.sql.ResultSet for JDK 1.0.2).
*
* No interface types are defined for result set handling.
* When an operation returns a result set, the entire result set
* must be transmitted from server to client before the client can begin
* processing the results.

```

```

*
* This approach has the potential to be much more efficient in a wide-area
* network environment than the use of interface types, which would result in a
* large number of small requests from client to server to fetch an entire
* result set. It is also preferable to streaming the result set into an
* octet sequence, since the IDL language mappings for the ResultSet
* type allow for convenient manipulation of result set data and meta-data.
*
* If a client requires the ability to process some results before all the
* results have been transmitted, the result set can be explicitly broken
* into batches by the server programmer using an IDL interface such as:
*
* interface SampleInterface
* {
*     TabularResults::ResultSet operationWhichReturnsResultSet(...);
*
*     TabularResults::ResultSet getMoreResults();
* };
*/
module TabularResults
{
    // The following constants define the column data types for result sets.
    // The actual constant values are equivalent to those in XOPEN (and JDBC).
    // The leading 'TYPE_' prefix is used since some of the type names are IDL
    // reserved words.
    const long TYPE_BIGINT      = -5;
    const long TYPE_BINARY     = -2;
    const long TYPE_BIT        = -7;
    const long TYPE_CHAR       = 1;
    const long TYPE_DATE       = 91;
    const long TYPE_DECIMAL    = 3;
    const long TYPE_DOUBLE     = 8;
    const long TYPE_FLOAT      = 6;
    const long TYPE_INTEGER    = 4;
    const long TYPE_LONGVARBINARY = -4;
    const long TYPE_LONGVARCHAR = -1;
    const long TYPE_NUMERIC    = 2;
    const long TYPE_REAL       = 7;
    const long TYPE_SMALLINT   = 5;
    const long TYPE_TIME       = 92;
    const long TYPE_TIMESTAMP  = 93;
    const long TYPE_TINYINT    = -6;
    const long TYPE_VARBINARY  = -3;
    const long TYPE_VARCHAR    = 12;
    // The following constants define flags for result columns. These are
    // used to convey meta-data which is not indicated by the column type.
    const unsigned long FLAG_AUTO_INCREMENT    = 1;
    const unsigned long FLAG_CASE_SENSITIVE   = 2;
    const unsigned long FLAG_CURRENCY         = 4;
    const unsigned long FLAG_NOT_NULLABLE     = 8;
    const unsigned long FLAG_NULLABLE        = 16;
    const unsigned long FLAG_READONLY         = 32;
    const unsigned long FLAG_SEARCHABLE       = 64;

```

```

const unsigned long FLAG_UNSIGNED          = 128;
const unsigned long FLAG_WRITABLE         = 256;
const unsigned long FLAG_DEFINITELY_WRITABLE = 512;
typedef sequence < boolean > BooleanSeq;
typedef sequence < octet > OctetSeq;
typedef sequence < short > ShortSeq;
typedef sequence < long > LongSeq;
typedef sequence < float > FloatSeq;
typedef sequence < double > DoubleSeq;
typedef sequence < string > StringSeq;
typedef sequence < BCD::Binary > BinarySeq;
typedef sequence < BCD::Decimal > DecimalSeq;
typedef sequence < MJD::Date > DateSeq;
typedef sequence < MJD::Time > TimeSeq;
typedef sequence < MJD::Timestamp > TimestampSeq;
/*
 * The Data type represents an entire column in a result set.
 * Data is stored in a result set in column-major order. This means the
 * column data type (the union discriminator) only needs to be transmitted
 * over the network once, and minimises padding when using GIOP.
 *
 * Notes:
 * Type BIGINT uses BCD::Decimal because some CORBA ORBs do not support
 * the IDL "long long" type that was introduced with CORBA 2.1 / GIOP 1.1.
 */
union Data switch (long)
{
    case TYPE_BIT          : BooleanSeq booleanValues;
    case TYPE_TINYINT      : OctetSeq octetValues;
    case TYPE_SMALLINT     : ShortSeq shortValues;
    case TYPE_INTEGER      : LongSeq longValues;
    case TYPE_REAL         : FloatSeq floatValues;
    case TYPE_DOUBLE       :
    case TYPE_FLOAT        : DoubleSeq doubleValues;
    case TYPE_CHAR         :
    case TYPE_LONGVARCHAR  :
    case TYPE_VARCHAR      : StringSeq stringValues;
    case TYPE_BINARY       :
    case TYPE_LONGVARBINARY :
    case TYPE_VARBINARY    : BinarySeq binaryValues;
    case TYPE_BIGINT       :
    case TYPE_DECIMAL      :
    case TYPE_NUMERIC      : DecimalSeq decimalValues;
    case TYPE_DATE         : DateSeq dateValues;
    case TYPE_TIME         : TimeSeq timeValues;
    case TYPE_TIMESTAMP    : TimestampSeq timestampValues;
};
/*
 * A result column consists of meta data and data values, as well as
 * a sequence indicating which rows contain null values. The length of
 * 'nulls' may be less than the number of result rows. The default value
 * if a row's 'nulls' entry is not present is false (i.e. non-null). This
 * optimisation is particularly useful when the column's 'flags' contains

```

```

* To avoid ambiguity the 'values' sequence must contain an entry for
* each row, in order. Where a row has a null value, there should
* nevertheless be a token entry in the 'values' sequence (which will
* be ignored). By default the entry for a null row in the values sequence
* will be the default for that number type.
* The precision for floating-point decimal values is calculated as
* 'width - sign - dot', where sign = 1 if the flags bit FLAG_UNSIGNED
* is set (otherwise sign = 0), and dot = 1 if scale > 0 (otherwise
* dot = 0).
* The scale must be zero for TYPE_BIGINT.
*/
struct Column
{
    unsigned long flags; // Column's meta-data flags
    unsigned long width; // Column's normal display width in characters
    unsigned long scale; // Column's scale for fixed-point decimal values
    string name; // Column's name
    string label; // Column's label (may be empty if equal to name)
    Data values; // Data values for this column (for all rows)
    BooleanSeq nulls; // May be less than the number of rows
};
typedef sequence < Column > ColumnSeq;
/*
* The ResultSet type may be used as the return type
* for an operation returning a single result set.
*/
struct ResultSet
{
    unsigned long rows;
    ColumnSeq columns;
};
/*
* The ResultSets type may be used as the return type
* for an operation returning multiple result sets.
*/
typedef sequence<ResultSet> ResultSets;
};

```

F.8 UI/007, B-Utility.idl

```

/*
***
This file defines the interface to the utility functions
for UTM CORBA clients.
*/
module Utility
{
    interface UTMUtility
    {
        /*
        *
        * Get the CDB time.
        * Value is returned as the number of seconds from UTC epoch 1/1/1970 )

```

F Definitions of registered IDL scripts (Normative)

```
*  
*/  
long getUTCTime();  
};  
};
```

G Definitions of registered XML Schemas (Normative)

G.1 Introduction

- G.1.1 The UTMC XML schema for both content and transport is published as a zipped folder of XSD and associated files.
- G.1.2 A development site for the UTMC XML schema is provided at www.kizoom.com/utmc, by kind courtesy of Kizoom.

H Definitions of registered Other Objects (Normative)

H.1 There are no other objects currently specified.