INTERNATIONAL STANDARD

IEC 60870-6-802

Edition 2.1

2005-09

Edition 2:2002 consolidated with amendment 1:2005

Telecontrol equipment and systems –

Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Object models

Reference number
IEC 60870-6-802:2002+A1:2005(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site (www.iec.ch)**

- **Catalogue of IEC publications**

  The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

  This summary of recently issued publications (www.iec.ch/online_news/ justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

  If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

  Email: custserv@iec.ch
  Tel: +41 22 919 02 11
  Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 60870-6-802

Edition 2.1

2005-09

Edition 2:2002 consolidated with amendment 1:2005

Telecontrol equipment and systems –

Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Object models

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

TELECONTROL EQUIPMENT AND SYSTEMS –

Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Object models

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 60870-6-802 has been prepared by IEC technical committee 57: Power system control and associated communications.

This consolidated version of IEC 60870-6-802 consists of the second edition (2002) [documents 57/575/FDIS and 57/583/RVD] and its amendment 1 (2005) [documents 57/740/FDIS and 57/745/RVD].

The technical content is therefore identical to the base edition and its amendment and has been prepared for user convenience.

It bears the edition number 2.1.

A vertical line in the margin shows where the base publication has been modified by amendment 1.

Annex A is for information only.

The committee has decided that the contents of the base publication and its amendments will remain unchanged until the maintenance result date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

INTRODUCTION

The primary purpose of Telecontrol Application Service Element (TASE.2) is to transfer data between control systems and to initiate control actions. Data is represented by object instances. This part of IEC 60870 proposes object models from which to define object instances. The object models represent objects for transfer. The local system may not maintain a copy of every attribute of an object instance.

The object models presented herein are specific to "control centre" or "utility" operations and applications; objects required to implement the TASE.2 protocol and services are found in IEC 60870-6-503. Since needs will vary, the object models presented here provide only a base; extensions or additional models may be necessary for two systems to exchange data not defined within this standard.

It is by definition that the attribute values (i.e. data) are managed by the owner (i.e. source) of an object instance. The method of acquiring the values are implementation dependent; therefore accuracy is a local matter.

The notation of the object modelling used for the objects specified in clause 5 is defined in IEC 60870-6-503. It should be noted that this part of IEC 60870 is based on the TASE.2 services and protocol. To understand the modelling and semantics of this standard, some basic knowledge of IEC 60870-6-503 is recommended.

Clause 5 describes the control centre-specific object models and their application. They are intended to provide information to explain the function of the data.

Clause 6 defines a set of MMS type descriptions for use in exchanging the values of instances of the defined object models. It is important to note that not all attributes of the object models are mapped to types. Some attributes are described simply to define the processing required by the owner of the data and are never exchanged between control centres. Other attributes are used to determine the specific types of MMS variables used for the mapping, and therefore do not appear as exchanged values themselves. A single object model may also be mapped onto several distinct MMS variables, based on the type of access and the TASE.2 services required.

Clause 7 describes the mapping of instances of each object type MMS variables and named variable lists for implementing the exchange.

Clause 8 describes device-specific codes and semantics to be used with the general objects.

An informative annex is included which describes some typical interchange scheduling scenarios, along with the use of TASE.2 objects to implement the schedule exchange.

TELECONTROL EQUIPMENT AND SYSTEMS –

Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Object models

1 Scope

This part of IEC 60870 specifies a method of exchanging time-critical control centre data through wide-area and local-area networks using a full ISO compliant protocol stack. It contains provisions for supporting both centralized and distributed architectures. The standard includes the exchange of real-time data indications, control operations, time series data, scheduling and accounting information, remote program control and event notification.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60870-5-101:1995, *Telecontrol equipment and systems – Part 5: Transmission protocols – Section 101: Companion standard for basic telecontrol tasks*

IEC 60870-6-503:2002, *Telecontrol equipment and systems – Part 6: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – Section 503: TASE.2 Services and protocol*

ISO 9506-1:2000, *Industrial automation systems – Manufacturing message specification – Part 1: Service definition*

ISO 9506-2:2000, *Industrial automation systems – Manufacturing message specification – Part 2: Protocol specification*

3 Definitions

For the purposes of this part of IEC 60870, the definitions in the above referenced standards apply.

4 Abbreviations

For the purposes of this part of IEC 60870, all the abbreviations defined in the above referenced standards apply.

5 Object models

Object models are required for various functions within a system. This clause delineates abstract object models based on functionality. Object models within one functional area may be used in another functional area.

5.1 Supervisory Control and Data Acquisition

The object models in this clause are derived from the historical perspective of Supervisory Control and Data Acquisition (SCADA) systems. The following text presents the context within which the object models are defined.

Fundamental to SCADA systems are two key functions: control and indication. The control function is associated with the output of data whereas the indication function is associated with the input of data. A more recent concept that is finding usage is the control and indication function where data output may also be input (i.e. bi-directional).

The previous identified functions within SCADA systems are mapped to point equipment (point). The primary attribute of a point is the data value. SCADA systems define three types of data for points: analog, digital and state.

The association of one or more points together is used to represent devices. For example, a breaker device may be represented by a control point and an indication point. The control point represents the new state that one desires for the breaker device. The indication point represents the current state of the breaker device. For SCADA to SCADA data exchange (e.g. control centre to control centre, control centre to SCADA master, etc.), additional data is often associated with point data. Quality of point data is often exchanged to defined whether the data is valid or not. In addition, for data that may be updated from alternate sources, quality often identifies the alternate source. Select-Before-Operate control is associated with Control Points for momentary inhibiting access except from one source. Two other informative data values are: time stamp and change of value counter. The time stamp, when available, details when a data value last changed. The change of value counter, when available, details the number of changes to the value.

From the context presented the primary object models required are: Indication Point, and Control Point. The attributes Point Value, Quality, Select-Before-Operate, Time Stamp, and Change of Value Counter are required to meet the desired functionality for data exchange. The Indication Point and Control Point models may be logically combined to a single model to represent a device which implements a control function with a status indication as to its success/failure. The combined logical model will result in the same logical attributes, and map onto the same MMS types as the independent models.

5.1.1 IndicationPoint Object

An IndicationPoint object represents an actual input point.

Object: **IndicationPoint** (Read Only)
- Key Attribute: PointName
- Attribute: PointType (REAL, STATE, DISCRETE)
- Constraint PointType=REAL
  - Attribute: PointRealValue
- Constraint PointType=STATE
  - Attribute:PointStateValue
- Constraint PointType=DISCRETE
  - Attribute: PointDiscreteValue
- Attribute: QualityClass: (QUALITY, NOQUALITY)

Constraint: QualityClass = QUALITY
Attribute: Validity (VALID, HELD, SUSPECT, NOTVALID)
Attribute: CurrentSource (TELEMETERED, CALCULATED, ENTERED, ESTIMATED)
Attribute: NormalSource (TELEMETERED, CALCULATED, ENTERED, ESTIMATED)
Attribute: NormalValue (NORMAL,ABNORMAL)
Attribute: TimeStampClass: (TIMESTAMP, TIMESTAMPEXTENDED, NOTIMESTAMP)
Constraint: TimeStampClass = TIMESTAMP
Attribute: TimeStamp
Attribute: TimeStampQuality: (VALID, INVALID)
Constraint: TimeStampClass = TIMESTAMPEXTENDED
Attribute: TimeStampExtended
Attribute: TimeStampQuality: (VALID, INVALID)
Attribute: COVClass: (COV, NOCOV)
Constraint: COVClass = COV
Attribute: COVCounter

PointName
The PointName attribute uniquely identifies the object.

PointType
The PointType attribute identifies the type of input point, and must be one of the following: REAL, STATE, DISCRETE.

PointRealValue
The current value of the IndicationPoint, if the PointType attribute is REAL.

PointStateValue
The current value of the IndicationPoint, if the PointType attribute is STATE.

PointDiscreteValue
The current value of the IndicationPoint, if the PointType attribute is DISCRETE.

QualityClass
The QualityClass has the value QUALITY if the object instance has any of the quality attributes (Validity, CurrentSource, or NormalValue), and takes the value NOQUALITY if none of the attributes are present.

Validity
The Validity attribute specifies the validity or quality of the PointValue data it is associated with. These are based on the source system's interpretation as follows:

Validity	Description
VALID	Data value is valid
HELD	Previous data value has been held over. Interpretation is local
SUSPECT	Data value is questionable. Interpretation is local
NOTVALID	Data value is not valid

CurrentSource
The CurrentSource attribute specifies the current source of the PointValue data it is associated with as follows:

CurrentSource	Description
TELEMETERED	The data value was received from a telemetered site
CALCULATED	The data value was calculated based on other data values
ENTERED	The data value was entered manually
ESTIMATED	The data value was estimated (State Estimator, etc.)

NormalSource
The NormalSource attribute specifies the normal source of the PointValue data it is associated with as follows:

NormalSource	Description
TELEMETERED	The data value is normally received from a telemetered site
CALCULATED	The data value is normally calculated based on other data values
ENTERED	The data value is normally entered manually
ESTIMATED	The data value is normally estimated (State Estimator, etc.)

NormalValue
The NormalValue attribute reports whether value of the PointValue attribute is normal. Only one bit is set, it is defined as follows:

NormalValue	Description
NORMAL	The point value is that which has been configured as normal for the point
ABNORMAL	The point value is not that which has been configured as normal for the point

TimeStampClass
The TimeStampClass attribute has the value TIMESTAMP or TIMESTAMPEXTENDED if the IndicationPoint is time stamped, and has the value NOTIMESTAMP if the IndicationPoint contains no TimeStamp attribute.

TimeStamp
The TimeStamp attribute provides a time stamp (with a minimum resolution of one second) of when the value (attribute PointRealValue, PointStateValue or PointDiscreteValue) of the IndicationPoint was last changed. It is set at the earliest possible time after collection of the IndicationPoint value from the end device.

TimeStampExtended
The TimeStampExtended attribute provides a time stamp (with a resolution of one millisecond) of when the value (attribute PointRealValue, PointStateValue or PointDiscreteValue) of the IndicationPoint was last changed. It is set at the earliest possible time after collection of the IndicationPoint value from the end device.

TimeStampQuality
The TimeStampQuality attribute has the value VALID if the current value of the TimeStamp attribute contains the time stamp of when the value was last changed, and has the value INVALID at all other times.

COVClass
The COVClass (**C**hange **O**f **V**alue Counter) attribute has the value COV if the IndicationPoint contains a COVCounter attribute, otherwise it has the value NOCOV.

COVCounter
The COVCounter attribute specifies the number of times the value (attribute PointRealValue, PointStateValue, or PointDiscreteValue) of the IndicationPoint has changed. It is incremented each time the owner sets a new value for the IndicationPoint.

5.1.2 ControlPoint Object

A ControlPoint Object is an integral part of the services provided by TASE.2. It is used to represent values of various types of data typical of SCADA and energy management systems. Typically, a ControlPoint object will be associated with some real world object.

Object: **ControlPoint** (Write Only, except for attributes CheckBackName, Tag, State and Reason)
- Key Attribute: ControlPointName
- Attribute: ControlPointType: (COMMAND, SETPOINT)
- Constraint: ControlPointType = COMMAND
  - Attribute: CommandValue
- Constraint: ControlPointType = SETPOINT
  - Attribute: SetPointType: (REAL, DISCRETE)
  - Constraint SetpointType=REAL
    - Attribute: SetpointRealValue
  - Constraint SetpointType=DISCRETE
    - Attribute: SetpointDiscreteValue
- Attribute: DeviceClass: (SBO, NONSBO)
- Constraint: DeviceClass = SBO
  - Attribute: CheckBackName
  - Attribute: State: (SELECTED, NOTSELECTED)
  - Attribute: Timeout
- Attribute: TagClass: (TAGGABLE, NONTAGGABLE)
- Constraint: TagClass = TAGGABLE
  - Attribute: Tag: (NO-TAG, OPEN-AND-CLOSE-INHIBIT, CLOSE-ONLY-INHIBIT)
  - Attribute: State: (IDLE, ARMED)
  - Attribute: Reason

ControlPointName
The ControlPointName attribute uniquely identifies the object.

ControlPointType
The value of the ControlPointType attribute for an instance of a ControlPoint will be COMMAND or SETPOINT, indicating the type of controlled device.

CommandValue
The CommandValue attribute indicates the command for a device.

SetPointType
The value of the SetPointType attribute for an instance of a ControlPoint of ControlPointType SETPOINT will be REAL or DISCRETE, indicating the type of setpoint.

SetPointRealValue
The SetPointRealValue attribute may be set with the floating point value requested for the setpoint control.

SetPointDiscreteValue
The SetPointDiscreteValue attribute may be set with the integer value requested for the setpoint control.

DeviceClass
The DeviceClass attribute of an instance of a ControlPoint has the value SBO if the device requires a Select operation before being operated, and the value NONSBO otherwise.

CheckBackName
The CheckBackName attribute contains a symbolic description of the physical object being controlled. This data is returned by the system operating the physical object to the system requesting the operation so that the person or system requesting the operation can be assured the proper object has been selected.

State
The State attribute indicates whether the ControlPoint is SELECTED or NOTSELECTED.

Timeout
The Timeout attribute of an instance of a ControlPoint has the value of the maximum allowed time for which the ControlPoint of DeviceClass SBO may remain SELECTED before operation.

TagClass
The TagClass attribute of an instance of a ControlPoint has the value TAGGABLE if the instance contains a Tag attribute, and otherwise has the value NONTAGGABLE.

Tag
The Tag attribute indicates whether or not the ControlPoint is tagged, and if it is, what the level of tagging is. The Tag attribute can take on the values NO-TAG, OPEN-AND-CLOSE-INHIBIT, CLOSE-ONLY-INHIBIT.

Reason
The Reason attribute contains a message that indicates the reason for tagging.

5.1.3 Protection Equipment Event Object Model

The following object model represents the events generated in the operation of protection equipment. Start events are generated by the protection equipment when it detects faults. Trip events report commands to output circuits which are generated by the protection equipment when it decides to trip the circuit-breaker. Both events are transient information. The protection event models are based on IEC 60870-5-101.

Object: **ProtectionEvent**
- KeyAttribute: Name
- Attribute: ElapsedTimeValidity (VALID, INVALID)
- Attribute: Blocked (NOTBLOCKED, BLOCKED)
- Attribute: Substituted (NOTSUBSTITUTED, SUBSTITUTED)
- Attribute: Topical (TOPICAL, NOTTOPICAL)
- Attribute: EventValidity (VALID, INVALID)
- Attribute: ProtectionClass (SINGLE, PACKED)
- Constraint: ProtectionClass = SINGLE
  - Attribute: EventState (INDETERMINATE, OFF, ON)
  - Attribute: EventDuration
  - Attribute: EventTime
- Constraint: ProtectionClass = PACKED
  - Attribute: EventClass (START, TRIP)
  - Constraint: EventClass = START
    - Attribute: StartGeneral (NOSTART, START)
    - Attribute: StartPhase1 (NOSTART, START)
    - Attribute: StartPhase2 (NOSTART, START)
    - Attribute: StartPhase3 (NOSTART, START)
    - Attribute: StartEarth (NOSTART, START)
    - Attribute: StartReverse (NOSTART, START)
    - Attribute: DurationTime
    - Attribute: StartTime

Constraint: EventClass = TRIP
Attribute: TripGeneral (NOTRIP, TRIP)
Attribute: TripPhase1 (NOTRIP, TRIP)
Attribute: TripPhase2 (NOTRIP, TRIP)
Attribute: TripPhase3 (NOTRIP, TRIP)
Attribute: OperatingTime
Attribute: TripTime

Name
The Name attribute uniquely identifies the protection event.

ElapsedTimeValidity
The elapsed time (attribute EventDuration, DurationTime, or OperatingTime depending on the event type) is valid if it was correctly acquired. If the acquisition function detects invalid conditions, the ElapsedTimeValidity attribute is INVALID, otherwise it is VALID.

Blocked
The Blocked attribute is BLOCKED if the value of protection event is blocked for transmission, and is NOTBLOCKED otherwise. The value remains in the state that was acquired before it was blocked. Blocking and deblocking may be initiated by a local lock or by a local automatic cause.

Substituted
The Substituted attribute takes the value SUBSTITUTED if the event was provided by input of an operator (dispatcher) or by an automated source.

Topical
The Topical attribute is TOPICAL if the most recent update was successful, and is NOTTOPICAL if it was not updated successfully during a specified time interval or is unavailable.

EventValidity
The EventValidity attribute takes the value INVALID if the acquisition function recognizes abnormal conditions of the information source, otherwise it is VALID.

ProtectionClass
The ProtectionClass attribute identifies the type of protection event, and must be one of the following: SINGLE or PACKED.

EventState
The EventState attribute of a SINGLE protection event takes the value of the protection event: OFF, ON or INDETERMINATE.

EventDuration
The EventDuration attribute takes the value of the event duration (total time the fault was detected) or operation time (time between start of operation and trip command execution).

EventTime
The EventTime attribute signifies the time of the start of the operation.

EventClass
The type of protection event being reported. The value START signifies a start event, and TRIP signifies a trip event.

StartGeneral
The value NOSTART signifies no general start of operation, and START signifies that the event includes a general start of operation.

StartPhase1
The value NOSTART for StartPhase1 implies that Phase L1 was not involved in the event, START implies that it was involved.

StartPhase2
The value NOSTART for StartPhase2 implies that Phase L2 was not involved in the event, START implies that it was involved.

StartPhase3
The value NOSTART for StartPhase3 implies that Phase L3 was not involved in the event, START implies that it was involved.

StartEarth
The value NOSTART for StartEarth implies that earth current was not involved in the event, START implies that it was involved.

StartReverse
The value NOSTART for StartReverse implies that reverse direction was not involved in the event, START implies that it was involved.

DurationTime
Time in milliseconds from the start of operation until the end of operation.

StartTime
Time of the start of operation of the protection equipment.

TripGeneral
The TripGeneral attribute takes on the value of TRIP if a general command to the output circuit was issued during the operation, NOTRIP otherwise.

TripPhase1
The TripPhase1 attribute takes on the value of TRIP if a command to output circuit Phase L1 command was issued during the operation, NOTRIP otherwise.

TripPhase2
The TripPhase2 attribute takes on the value of TRIP if a command to output circuit Phase L2 command was issued during the operation, NOTRIP otherwise.

TripPhase3
The TripPhase3 attribute takes on the value of TRIP if a command to output circuit Phase L3 command was issued during the operation, NOTRIP otherwise.

OperatingTime
The time in milliseconds from the start of operation until the first command to an output circuit was issued.

TripTime
Time of the start of the operation.

5.2 Transfer Accounts

One of the key control centre application requirements is the ability to exchange "scheduling" and "accounting" information. In the utility world, "schedules" is a term that generally means an amount of electrical energy transferred from one system to another on a periodic basis for a certain interval of time under the restrictions of a formal agreement. From a data exchange standpoint, exchanging "schedules" has been expanded to include the exchange of any periodic or profile data for control centre energy scheduling, accounting or monitoring applications. Thus exchanging schedules may also mean exchanging generation, actual interchange, loads, price information, memo accounts, etc. Again, any information may be exchanged if it can be modelled as periodic or profile data.

Schedule accounts and accounting information accounts specify a list of quantities (energy, capacity or some other user-defined quantity) to be exchanged during sequential time periods. These transactions may be specified either as flat periodic values or as ramped profiles. In addition, schedule accounts may contain a list of transmission segments defining the paths used to implement the transaction.

The subclauses below discuss the object models for transfer accounts.

5.2.1 TransferAccount Object

A Transfer Account object represents what, where, when, and how much is transferred between two utilities in a particular account. It may also represent generation schedules and other energy delivery schedules within a utility. It is a container for a number of different attributes and objects which together define the entire transfer account definition, i.e. which account, when is the effective time frame, and what are the periodic or profile values of the data. Refer to annex A for more information concerning how these object definitions relate to each other.

Object: **TransferAccount**

- Key Attribute: TransferAccountReference
- Attribute: SendUtility
- Attribute: ReceiveUtility
- Attribute: SellingAgent
- Attribute: BuyingAgent
- Attribute: TimeStamp
- Attribute: TransactionCode
- Attribute: NumberOfLocalReference
- Attribute: ListOfLocalReference
- Attribute: Name
- Attribute: TransmissionSegmentOption (INCLUDED, NOTINCLUDED)
- Constraint: TransmissionSegmentOption=INCLUDED
  - Attribute: NumberOfTransSegments
  - Attribute: ListOfTransmissionSegment
- Attribute: DataType (PERIODIC, PROFILE)
- Constraint: DataType = PERIODIC
  - Attribute: StartTime
  - Attribute: PeriodResolution
  - Attribute: NumberOfPeriods
  - Attribute: ListOfPeriodicValues
- Constraint: DataType = PROFILE
- Attribute: NumberOfProfiles
- Attribute: ListOfProfileValues

TransferAccountReference
The TransferAccountReference attribute specifies a unique reference value between the sender and the receiver to identify this particular transfer account.

SendUtility
The SendUtility attribute specifies the utility initially delivering the energy or service.

ReceiveUtility
The ReceiveUtility attribute specifies the utility ultimately receiving the energy or service.

SellingAgent
Currently, the seller is almost always the same as SendUtility. However, as a result of deregulation and open access, it is likely in the future that the seller may be an independent power producer within the area serviced by the SendUtility. Obviously, in such cases it is necessary to distinguish between the seller and the sending utility; this attribute allows such a distinction.

BuyingAgent
Currently, the buyer is almost always the same as ReceiveUtility. However, as a result of deregulation and open access, it is likely in the future that the buyer may be a municipality or cooperative within the area serviced by the ReceiveUtility. Obviously, in such cases it is necessary to distinguish between the buyer and the receiving utility; this attribute allows such a distinction.

TimeStamp
The TimeStamp attribute provides a means for the receiver to know when the sender sent this message. The time stamp indicates when the application created the Transfer Account data for the purpose of transmitting the data. It is not related to the time to which the Transfer Account data itself applies. If the applications providing data to TASE.2 include a transmission (or object creation) time stamp, that time stamp is used. Otherwise, TASE.2 provides the time stamp prior to passing the Transfer Account object to MMS.

TransactionCode
The TransactionCode attribute provides status on the TransferAccount transaction. The values are shown below:

TransactionCode	Description
NEW	Original submittal of a newly proposed schedule or of a report
REQUESTED	Report or schedule is requested via the Account Request operation
REVISED	Revised version of previously sent schedule or report
CONFIRMED	Schedule or report has now been scheduled for actual implementation and is not subject to further approval cycles. A *curtailment, halt, revision,* or *deletion* after this stage requires a new schedule or report to be sent
DELETED	Any time the originator of the schedule or report cancels the transaction (for example, due to equipment failure)

NumberOfLocalReference
This attribute contains the count of entries in the ListOfLocalReference.

ListOfLocalReference
The ListOfLocalReference contains locally defined parameters that specify how the Transfer Account Object is to be treated as a whole. For example, it might specify that the schedule or transaction applies to all Mondays during Summer. The meaning of the parameter list is a local matter and is agreed upon among the participating parties for each type of Transfer Account object, for all Transfer Account objects, or for a combination of the two as needed.

Name
The Name attribute provides an option for the user application to include string related information associated with the Transfer Account object. For example, a schedule may need to identify a generating unit by name. Whether or not this attribute is used and its semantics are a local implementation matter.

TransmissionSegmentOption
The TransmissionSegmentOption indicates whether or not a ListOfTransmissionSegments is included in the Transfer Account object. Transmission segments may be included in a Transfer Account object independently of the kind of data the Transfer Account object represents, i.e. it does not have to be included only for energy buy-sell transactions between utilities where wheeling is involved. Transmission segments can be used in any Transfer Account object to convey any information between multiple parties when some of the information is unique to each party and it is desired to provide all of the information in the account to all parties.

NumberOfTransSegments
This attribute defines the number of transmission segments to be included in the report.

ListOfTransmissionSegments
The ListOfTransmissionSegments attribute describes the various transmission paths (or segments) that an interchange schedule may take to get from the SendUtility to the ReceiveUtility, or to describe other information that is significant to intermediate parties of any transaction. Each TransmissionSegment object provided describes one component of the path, or data for one intermediate party. As many or as few TransmissionSegment Objects as required may be added to the list.

DataType
The data contained in a Transfer Account object can be periodic as is the case of most of the data for generation schedules, energy transactions, and billing. The data can also define a series of ramping requirements which together result in an energy (or price) profile. The DataType attribute indicates which of these two types of data the Transfer Account object contains. Note that a Transfer Account object cannot contain both periodic and profile data. However, there is no restriction on the use of the TransferAccountReference attribute with respect to the object's data type. Therefore, two Transfer Account objects could be transmitted with the same TransferAccountReference; the first would convey periodic type information while the other would convey profile type data.

StartTime
For a Transfer Account object containing periodic data, StartTime specifies the UTC time to which the first time period in the sequence applies.

PeriodResolution
For a Transfer Account object containing periodic data, PeriodResolution specifies the quantity of time to which each entry in the sequence applies.

NumberOfPeriods
This attribute defines the number of time periods being specified in the sequence defined by the Transfer Account object. Together with StartTime and PeriodResolution, it also defines the maximum time frame covered by the Transfer Account.

ListOfPeriodicValues
This attribute contains the bulk of the data associated with the account. For energy transactions, this is the data related to the end-use of the transaction that is, the transaction between the buyer and seller. Information specific to the wheeling partners of each transmission segment is specified in the ListOfTransmissionSegment attribute (if included).

NumberOfProfiles
This attribute defines the number of time periods specified in the ListOfProfileValues.

ListOfProfileValues
The ListOfProfileValues attribute describes profile, or ramping information. The number of ProfileValue objects passed in the message implies the number of inflection point changes in the profile – one for each change.

5.2.2 TransmissionSegment Object

A TransmissionSegment object is used to represent the specifics of a wheeling arrangement where a utility allows another utility the use of its high-voltage transmission grid to transfer energy to a third utility. This object may also be used to convey any information to a utility involved in a multi-utility transaction. The TransmissionSegment may describe either where the energy is to be received and which utility it is coming from (INONLY) or where the energy is to be delivered and which utility is to receive it (OUTONLY). Also a TransmissionSegment object may describe both the in and out utilities and their associated receipt and delivery points (INOUT). Or if the energy is being scheduled directly between two utilities (DIRECT), it may only specify the point of interchange. In addition, the utility providing the service and the utility paying for the transmission service (wheeling) may be specified.

Object: **TransmissionSegment**
- Attribute: TransmissionReference
- Attribute: UtilWheeling
- Attribute: TransmissionSegType (INONLY, OUTONLY, INOUT, DIRECT)
- Constraint: TransmissionSegType = INONLY
  - Attribute: UtilIn
  - Attribute: InterchangePtIn
- Constraint: TransmissionSegType = OUTONLY
  - Attribute: UtilOut
  - Attribute: InterchangePtOut
- Constraint:TransmissionSegType = INOUT
  - Attribute: UtilIn
  - Attribute: InterchangePtIn
  - Attribute: UtilOut
  - Attribute: InterchangePtOut
- Constraint: TransmissionSegType = DIRECT
  - Attribute: InterchangePt
- Attribute: UtilPaying
- Attribute: ListOfSegmentData

TransmissionReference
The TransmissionReference attribute specifies a value agreed upon between the sender and receiver of the message that further describes the transmission segment (transmission agreement reference number, etc.).

UtilWheeling
The UtilWheeling attribute specifies the utility providing the transmission services (if any).

TransmissionSegType
The TransmissionSegType attribute specifies the type of data in the TransmissionSegment object as follows:

TransmissionSegType	Description
INONLY	The object only specifies which utility the energy is coming from and where it is being received. This object type shall be used when the outbound side of the segment is either not known or not important
OUTONLY	The object only specifies which utility the energy is delivered to and where it is delivered. This object type shall be used when the inbound side of the segment is either not known or not important
INOUT	The object specifies both the in and out utilities and the associated receiving and delivering interchange points
DIRECT	The object specifies only the interchange point between the initial sending utility and the ultimate receiving utility. This object type shall be used when no intermediate utilities are providing transmission services

UtilIn
The UtilIn attribute specifies which utility the inbound energy is coming from.

InterchangePtIn
The InterchangePtIn attribute specifies which interchange point or interface the inbound energy is to be received from UtilIn.

UtilOut
The UtilOut attribute specifies which utility the outbound energy is delivered to.

InterchangePtOut
The InterchangePtOut attribute specifies which interchange point or interface the outbound energy is to be delivered to UtilOut.

InterchangePt
The InterchangePt attribute specifies which interchange point or interface the direct transfer of energy is to take place between SendUtility and RecvUtility.

UtilPaying
The UtilPaying attribute specifies which utility should be billed for any transmission service charges (wheeling, losses, etc.)

ListOfSegmentData
The ListOfSegmentData contains the information specific to the wheeling partners of each transmission segment. This may include data such as the energy transferred through the segment and/or the cost.

5.2.3 ProfileValue Object

A ProfileValue object represents a profile of a quantity or value being exchanged over time. The value may be energy, capacity, price and/or other information. It is represented by specifying a target value, ramp start time and a ramp rate. When a ProfileValue object is received which effects the magnitude of the exchange, the profile moves in the direction of the new target, starting at the time specified by the RampStartTime and at a rate specified by the RampDuration. If a zero is specified for the RampDuration (or it is not passed), the value does not begin to ramp but is implemented instantly at RampStartTime. Upon reaching the target value, it proceeds at a constant level through time. This constant level may be altered or terminated with another ProfileValue object.

Object: **ProfileValue**

Attribute: RampStartTime
Attribute: RampDuration
Attribute: ProfilePrice
Attribute: TargetClass (ENERGY, CAPACITY, OTHER)
Constraint: TargetClass = ENERGY
Attribute: ProfileEnergy
Constraint: TargetClass = CAPACITY
Attribute: ProfileCapacity
Constraint: TargetClass = OTHER
Attribute: ProfileOther

RampStartTime
The time when the change is to occur.

RampDuration
The length of time that the ramp transition may occur. The units are as specified in the bilateral agreements.

ProfilePrice
The price value is generally expressed in currency units per unit of measure (such as $/MWH) as defined in bilateral agreements.

TargetClass
The TargetClass attribute determines the type of profile. It takes on the values ENERGY, CAPACITY or OTHER.

ProfileEnergy
The energy value is generally expressed in energy/hours (such as MWH) as defined in bilateral agreements. This attribute exists if the TargetClass attribute is ENERGY.

ProfileCapacity
The capacity value, is generally expressed as a quantity of capacity (such as MW) as defined in bilateral agreements. This attribute exists if the TargetClass attribute is CAPACITY.

ProfileOther
Some other unspecified value as defined in bilateral agreements. Use local interpretation. This attribute exists if the TargetClass attribute is OTHER.

5.2.4 AccountRequest Object

An AccountRequest object is used to request account information. This request may be used for interchange schedule accounts and/or accounting information accounts. This object, when written, will cause the TASE.2 server to generate a TASE.2 Transfer Account Transfer Report (see IEC 60870-6-503) of the requested data. The Transfer Report will contain the identifier AccountRequested as the MMS variable being reported.

Object: **AccountRequest**

Key Attribute: AccountRequestName
Attribute: Transfer Account Reference
Attribute: StartTime
Attribute: Duration
Attribute: RequestId
Attribute: TaConditionRequested

AccountRequestName
The AccountRequestName uniquely identifies the AccountRequest object.

TransferAccountReference
The TransferAccountReference attribute specifies a unique reference value between the sender and the receiver to identify this particular transfer account.

StartTime
The StartTime attribute specifies the starting time that is being requested. This time is specified in seconds.

Duration
The duration attribute specifies the ending time that is being requested. It designates the number of seconds from StartTime.

RequestId
Used to match the response with the request. The value is returned in the Transfer Account report.

TaConditionsRequested
Identifies the type of data being requested.

5.3 Device Outage Object

A DeviceOutage object is used to communicate schedule information regarding device outages. It is composed of a number of objects which define the device which will be (was) affected and describe the time period for which the outage will occur.

Object: **DeviceOutage**
- Key Attribute: OutageReferenceId
- Attribute: OwningUtilityID
- Attribute: Timestamp
- Attribute: StationName
- Attribute: DeviceType (GENERATOR, TRANSFORMER, CAPACITOR, TRANSMISSION_CIRCUIT, BREAKER_SWITCH, INDUCTOR, OTHER)
- Attribute: DeviceName
- Attribute: DeviceNumber
- Attribute: DeviceRating
- Attribute: ActivityDateAndTime
- Attribute: Activity (NEWPLAN, REVISE, CANCEL, ACTUAL)
- Constraint: Activity = NEWPLAN, REVISE
  - Attribute: PlanType (SCHEDULED, ESTIMATED)
  - Attribute: PlannedOpenOrOutOfServiceDateAndTime
  - Attribute: PlannedCloseOrInServiceDateAndTime
  - Attribute: OutagePeriod (CONTINUOUS, DAILY, WEEKDAYS, OTHER)
  - Attribute: OutageType (FORCED, MAINTENANCE, PARTIAL, ECONOMY, UNPLANNED, OTHER)
  - Attribute: OutageAmountType (PARTIAL, FULL)
  - Constraint: OutageAmountType = PARTIAL
    - Attribute: Amount
    - Attribute: UpperOperatingLimit
    - Attribute: LowerOperatingLimit
  - Attribute: Class (INSERVICE, OUTSERVICE)

Constraint: Activity = ACTUAL
Attribute: Action (TRIPPED, OFFLINE, ONLINE, OPEN, CLOSE)
Constraint: Action = TRIPPED, OFFLINE, OPEN
Attribute: Affected Amount
Attribute: Comments
Attribute: OutageEffect

OutageReferenceId
The OutageReferenceId attribute is a unique reference value assigned by the originator for identifying this particular outage.

OwningUtilityID
ID of the utility or control area owning the equipment.

TimeStamp
The TimeStamp attribute provides the time at which a DeviceOutage object was generated.

StationName
Name of station at which the affected equipment, circuit, etc. resides.

DeviceType
Type of the device which is affected.

DeviceName
Name of the device which is affected.

DeviceNumber
The DeviceNumber attribute provides further qualification of DeviceName in cases where DeviceName may not provide sufficient specificity. As an example, multiple transmission lines may connect the same two substations and be assigned a single DeviceName. To uniquely identify one of these multiple lines, each line is assigned a DeviceNumber.

DeviceRating
The Device rating in KV, MW, MVAR.

ActivityDateAndTime
This is the time that the activity occurred or, for a planned activity or cancellation, this the time the activity is planned to occur.

Activity
This describes the type of activity reported. A NEWPLAN or REVISE requires dates for starting and ending the outage. A CANCEL only requires a cancellation date, which is recorded in the ActivityDateAndTime. An ACTUAL activity requires a date the event occurred, which is recorded in the ActivityDateAndTime, and a description of the actual event that occurred.

PlanType
A plan can be either a scheduled activity (with firm committed dates) or an activity with only estimated dates.

PlannedOpenOrOutOfServiceDateAndTime
Date and time the device is going to be taken out of service (or switch/breaker is to be opened).

PlannedCloseOrInServiceDateAndTime
Date and time the device is going to be returned to service (or switch/breaker is to be closed).

OutagePeriod
This describes the periodicity of the outage for plans which are to be executed periodically, such as daily or weekly.

OutageType
This describes the reason the equipment is being taken out of service. FORCED is a controlled but unscheduled outage MAINTENANCE is a scheduled outage for maintenance purposes PARTIAL is an outage where only a portion of the capacity is removed from service. ECONOMY is an outage planned for economic reasons. UNPLANNED is an unscheduled outage.

OutageAmountType
This describes whether the equipment outage is PARTIAL, in that some capacity is still available, or FULL, in which case no capacity is available.

Amount
For partial outages, this is the amount of normal capacity which will be unavailable during outage.

UpperOperatingLimit
Upper limit of operating range during outage.

LowerOperatingLimit
Lower limit of operating range during outage.

Class
Specifies whether device is in service or out of service when outage occurs.

Action
Describes the event which caused the outage or return to service.

AffectedAmount
This describes the load being carried at the time the outage starts.

Comments
This field contains 128 characters of ASCII text which may be used for operator comments.

OutageEffect
Effect of outage on transfer capacity. The field contains up to 128 characters of ASCII text which may be used to describe the effect of the outage on transfer capacity.

5.4 InformationBuffer Object

An InformationBuffer object is used to send multiple line ASCII text messages or binary data. It may be used to transfer messages limited in size to the maximum message size of the underlying communications structure (i.e. the maximum MMS PDU size). The application and coding of the content of this object is outside the scope of TASE.2, and is left as a local issue for agreement between the sending and receiving implementations. Note that this object is referred to as the Information Message object in IEC 60870-6-503.

Object: **InformationBuffer**

Key Attribute: InfoReference
Attribute: LocalReference
Attribute: MessageId
Attribute: Size
Attribute: InfoStream

InfoReference
The InfoReference attribute uniquely identifies the object. It is used to identify and/or trigger some special handling required by the receiving system.

LocalReference
The LocalReference attribute specifies a value agreed upon between the sender and receiver of the message that further identifies the message (file name, application identification, etc.).

MessageId
Identifies the particular instance of a message.

Size
The length of the valid data in the InfoStream.

InfoStream
The InfoStream attribute contains the byte stream of information being passed. It is limited only by the maximum size of a single message.

5.5 Power Plant Objects

These objects are intended for use with TASE.2 services, as defined in IEC 60870-6-503. The first two objects shall be reported using the same mechanism as TASE.2 schedule and accounting object model, possibly with redefined **Condition** codes. The third object may require a slightly different mechanism to be added to TASE.2 to allow for simple negotiation.

5.5.1 Availability Report Object

The following object represents a report from the plant to either a GCS or a control centre EMS declaring the overall availability of a unit for a given future time period. This report is based on the overall predicted availability of the plant resources. The unit may be available to operate in a variety of operating modes, each of which may involve a different price structure.

This report can also be used for scheduling an outage for various reasons, such as maintenance or testing. The outage is proposed by the DCS system, along with a range of time during which the operation may be initiated. Copies of this object can be repeatedly exchanged between the control centre and power plant until an actual start time is established.

Once a fixed schedule has been established and accepted for a maintenance or testing operation, no further exchanges are required unless either party wishes to cancel the operation or re-open the negotiation. In this case, the party requests the existing schedule be cancelled. If the operation must be re-negotiated, it may be re-proposed with a new start range.

Object: **Availability**
- Key attribute: AvailabilityReferenceID
- Attribute: Timestamp
- Attribute: PlantReferenceID
- Attribute: UnitID
- Attribute: ReportStatus (PROPOSED, CONFIRMED, CANCELLED)
- Attribute: StartDateAndTime
- Attribute: StopDateAndTime
- Attribute: Duration
- Attribute: Availability Status (AVAILABLE,UNAVAILABLE)

Constraint: AVAILABLE
Attribute: EconomicImpact (YES,NO)
Constraint: YES
Attribute: PriceImpact
Attribute: RampRateImpact (YES,NO)
Constraint: YES
Attribute: MaxRampRateUp
Attribute: MaxRampRateDown
Attribute: CapacityImpact (YES,NO)
Constraint: YES
Attribute: UnitCapacity (GROSS,NET,BOTH)
Constraint: GROSS
Attribute: GrossMaxCapacity
Attribute: GrossMinCapacity
Constraint: NET
Attribute: NetMaxCapacity
Attribute: NetMinCapacity
Constraint: BOTH
Attribute: GrossMaxCapacity
Attribute: GrossMinCapacity
Attribute: NetMaxCapacity
Attribute: NetMinCapacity
Attribute: TypeOfAvailability (STANDBY, ONLINE)
Constraint: STANDBY
Attribute: TimeToOnline
Constraint: ONLINE:
Attribute: LFC (YES,NO)
Constraint: YES
Attribute: Dispatchable
Attribute: Regulating
Attribute: Manually_Loaded
Constraint: NO
Attribute: ReasonForNoLFC (STARTUP,UNSTABLE)
Constraint: (UNAVAILABLE)
Attribute: ReasonForUnavailable (FORCED,SCHEDULED,TESTING)
Attribute: ProvidingReserve (YES,NO)
Attribute: Comment

AvailabilityReferenceID
A unique identifier to be used in subsequent references to the schedule when revising.

TimeStamp
The time the report is sent.

PlantReferenceID
Unique identifier for plant.

UnitID
Unique identifier for unit reported on.

ReportStatus
Status of availability report. PROPOSED indicates either the first or a revised schedule. If proposed, the start and stop date and time refer to the earliest start and stop times, with a duration stating the actual time estimated. CONFIRMED indicates a schedule accepted by the control centre, in which case the start and stop times refer to scheduled times. Duration may then be blank. CANCELLED is self explanatory.

StartDateAndTime
Either the earliest start date and time (for PROPOSED) or scheduled start date and time (for CONFIRMED).

StopDateAndTime
Either the earliest stop date and time (for PROPOSED) or scheduled stop date and time (for CONFIRMED).

Duration
The period of time covered by the report.

AvailabilityStatus
AVAILABLE indicates that the unit is able to generate power. UNAVAILABLE indicates the unit is offline and not available for scheduling.

EconomicImpact
If TRUE, indicates there is a price impact associated with this report.

PriceImpact
The absolute price associated with power generated by the unit for the period covered by the report.

RampRateImpact
If TRUE, indicates there is a ramp rate impact associated with the report.

MaxRampRateUp
The maximum predicted ramp rate up (MW) which will be attainable by the unit during the time period.

MaxRampRateDown
The maximum predicted ramp rate down (MW) which will be attainable by the unit during the time period.

CapacityImpact
If TRUE, indicates if there is an impact on capacity associated with this report.

UnitCapacity
Indicates if unit capacity reported is GROSS (direct output from unit), NET (net output onto line), or BOTH.

GrossMaxCapacity
The predicted gross maximum operating capacity (MW) of the unit for throughout the time period.

GrossMinCapacity
The predicted gross minimum operating capacity (MW) of the unit for throughout the time period.

NetMaxCapacity
The predicted net maximum operating capacity (MW) of the unit for throughout the time period.

NetMinCapacity
The predicted net minimum operating capacity (MW) of the unit for throughout the time period.

TypeOfAvailability
STANDBY indicates the unit is available to generate power but is not online. ONLINE indicates the unit is synchronized and generating power.

TimeToOnline
The time until the unit can be brought online.

LFC
If TRUE, indicates unit is available for control and which type of control. The LFC components below describe the availability of the unit for various types of control. Each type of use requires a distinct price (i.e., use as a regulating unit would require a different price than use as a base loaded unit). The components may be used in various combinations.

Dispatchable
Available to be operated to an externally generated setpoint.

Regulating
Available to be operated for the purpose of reducing Area Control Error (ACE) via LFC from a GCS or EMS.

Manually_Loaded
Available to be controlled locally.

ReasonForNoLFC
Reason why unit cannot be load frequency controlled. STARTUP indicates the unit is not yet up to desired operating point. UNSTABLE indicates the unit is at an operating point unsuitable for LFC.

ReasonForUnavailable
If the unit is out of service, the reason can be FORCED (controlled but unscheduled outage), SCHEDULED (for maintenance or other reasons), or TESTING.

ProvidingReserve
If YES, the unit is currently providing reserve.

Comment
Text string up to 256 characters to add any user defined description to the report.

5.5.2 Real Time Status Object

The following object represents a report from the plant to either a GCS or a control centre EMS declaring the overall operating mode of a unit at the time of the report. The real time report differs from the availability report in that:

- it defines the actual status of a unit, not a prediction;
- it includes the attributes which report external blocks, if any.

These real time reports may be issued at any time by the plant DCS system to report changes in the current status of a unit.

Object: **RealTimeStatus**
- Key Attribute: RealTimeStatusReferenceID
- Attribute: Timestamp
- Attribute: PlantReferenceID
- Attribute: UnitID
- Attribute: Availability Status (AVAILABLE,UNAVAILABLE)
- Constraint: AVAILABLE
  - Attribute: MaxRampRateUp
  - Attribute: MaxRampRateDown
  - Attribute: UnitCapacity (GROSS,NET,BOTH)
  - Constraint: GROSS
    - Attribute: GrossMaxCapacity
    - Attribute: GrossMinCapacity
  - Constraint: NET
    - Attribute: NetMaxCapacity
    - Attribute: NetMinCapacity
  - Constraint: BOTH
    - Attribute: GrossMaxCapacity
    - Attribute: GrossMinCapacity
    - Attribute: NetMaxCapacity
    - Attribute: NetMinCapacity
  - Attribute: TypeOfAvailability (STANDBY, ONLINE)
  - Constraint: STANDBY
    - Attribute: TimeToOnline
  - Constraint: ONLINE:
    - Attribute: LFC (YES,NO)
    - Constraint: YES
      - Attribute: Dispatched
      - Attribute: Regulating
      - Attribute: Manually_Loaded
    - Constraint: NO
      - Attribute: ReasonForNoLFC (STARTUP,UNSTABLE)
  - Attribute: ExternallyBlockedHigh (YES,NO)
  - Attribute: ExternallyBlockedLow (YES,NO)
- Constraint: UNAVAILABLE
  - Attribute: ReasonForUnavailable (FORCED,SCHEDULED,TESTING,EQUIPMENT)
- Attribute: ProvidingReserve (YES,NO)

RealTimeStatusReferenceID
A unique identifier to be used in subsequent references to the report.

TimeStamp
The time the report is sent.

PlantReferenceID
Unique identifier for plant.

UnitID
Unique identifier for unit reported on.

AvailabilityStatus
AVAILABLE indicates that the unit is able to generate power. UNAVAILABLE indicates the unit is offline and not available for scheduling.

MaxRampRateUp
The maximum predicted ramp rate up (MW) which will be attainable by the unit during the time period.

MaxRampRateDown
The maximum predicted ramp rate down (MW) which will be attainable by the unit during the time period.

UnitCapacity
Indicates if unit capacity reported is GROSS (direct output from unit), NET (net output onto line), or BOTH.

GrossMaxCapacity
The predicted gross maximum operating capacity (MW) of the unit for throughout the time period.

GrossMinCapacity
The predicted gross minimum operating capacity (MW) of the unit for throughout the time period.

NetMaxCapacity
The predicted net maximum operating capacity (MW) of the unit for throughout the time period.

NetMinCapacity
The predicted net minimum operating capacity (MW) of the unit for throughout the time period.

TypeOfAvailability
STANDBY indicates the unit is available to generate power but is not online. ONLINE indicates the unit is synchronized and generating power.

TimeToOnline
The time until the unit can be brought online.

LFC
If TRUE, indicates unit is under load frequency control and if so, which type of control. The LFC components below describe the various types of control. The components may be used in various combinations.

Dispatched
Operated to an externally generated setpoint.

Regulating
Operated for the purpose of reducing Area Control Error (ACE) via LFC from a GCS or EMS.

Manually_Loaded
Controlled locally.

ReasonForNoLFC
Reason why unit cannot be load frequency controlled. STARTUP indicates the unit is not yet up to desired operating point. UNSTABLE indicates the unit is at an operating point unsuitable for LFC.

ExternallyBlockedHigh
If TRUE, unit is temporarily unable to increase output.

ExternallyBlockedLow
If TRUE, unit is temporarily unable to decrease output.

ReasonForUnavailable
If the unit is out of service, the reason can be FORCED (controlled but unscheduled outage), SCHEDULED (for maintenance or other reasons), TESTING, or EQUIPMENT (such as due to an equipment failure).

ProvidingReserve
If YES, the unit is currently providing reserve.

5.5.3 Forecast Schedule Object

The following object model represents a forecast of intended scheduling from either a GCS or a control centre EMS to the plant. The schedule consists of a MW vs. time trend, as well as the expected operating mode (LFC_Component) over time. These forecasts may be either short term (over hours), or long term (over days).

The object model provides flexibility in what information is included with the schedule. A general purpose matrix object is used that provides for user-defined columns. It is expected that at least two columns will be used to provide Mw values and LFC mode of operation. The LFC modes expected are Manual, Dispatched, Regulating-Manual, and Regulating-Dispatched, but these are only examples. A code may be used to uniquely represent each mode. Each row represents an increment in time.

Object: **ForecastSchedule**
- Key Attribute: ForecastScheduleReferenceID
- Attribute: PlantReferenceID
- Attribute: UnitID
- Attribute: ForecastType (GENERATION,RESERVE,BOTH)
- Attribute: StartTime
- Attribute: PeriodResolution
- Attribute: NumberOfPeriods
- Attribute: ListOfForecasts

ForecastScheduleReferenceID
Unique identifier used for referring to the report.

PlantReferenceID
Unique identifier for plant.

UnitID
Unique identifier for unit reported on.

ForecastType
Describes the intended use of the unit. GENERATION indicates scheduling for base load. RESERVE indicates use only for reserve capacity.

StartTime
The time the forecast schedule begins.

PeriodResolution
The time increment between forecast values.

NumberOfPeriods
The number of forecast values.

ListOfForecasts
The scheduled Mw values for each time increment in the schedule along with the LFC mode of operation at each time increment.

5.5.4 Curve Object

The following object model represents a report from the plant to either a GCS or a control centre EMS of a new curve for computing such things as heat rate, MVAR capability, and cost. The method of generation of the curve is a local matter. The curve is represented as a sequence of curve segments, with each segment defined in terms of a polynomial. Each polynomial is in turn represented as a sequence of coefficients for each term in the polynomial.

Object: **CurveSegmentDescription**

Attribute: Order
Attribute: LowRange
Attribute: HighRange
Attribute: Sequence of Coefficients

Order
The order of the polynomial which represents the segment of the curve.

LowRange
The start of the curve interval which is represented by the segment. Note that if the LowRange is above the HighRange of the previous CurveSegmentDescription, the curve is undefined for the interval in between.

HighRange
The end of the curve interval which is represented by the segment. The **HighRange** must be greater than the **LowRange** of the segment.

Sequence of Coefficients
A list of floating point numbers, of length **Order**, corresponding to the coefficients of the polynomial which represents the segment of the curve. If the curve segment is represented by $A_0 + A_1x + A_2x^2 \ldots A_nx^n$, where n is the order of the polynomial, then the first element of the list represents A_0, and the last element in the list represents A_n.

Object: **Curve**

Key Attribute: CurveName
Attribute: PlantReferenceID
Attribute: UnitID
Attribute: CurveType (HEAT_RATE, IO, IHR, MVAR_CAP, COST, OPACITY, SOX, NOX, CO2, USER-DEFINED)
Attribute: NumberOfSegments
Attribute: Sequence of CurveSegmentDescription

CurveName
The unique name of the curve.

UnitID
The designator of the generation unit reporting a curve change.

CurveType
Identifies which of the standardized power plant curves is being reported. The full set of curve types remains to be determined. The initial set defined is: HEAT_RATE, IO, IHR, MVAR_CAP, COST, OPACITY, SOX, NOX, CO2, USER-DEFINED.

NumberOfSegments
The number of segments (curve intervals) which are required to represent the curve.

Sequence of CurveSegmentDescription
A list of segment descriptions, of length **NumberOfSegments**, which correspond to the polynomials representing each interval. The first segment is the lowest interval, the last segment is the highest interval.

5.5.5 Power System Dynamics Objects

The following additional data elements are required to support communications of the power system dynamics between the power plant and the GCS or control centre EMS system. These scalar quantities may be represented as simple TASE.2 Data Value objects.

Information Per Generating Unit

Cost

Transmission/Pool Security Centre

Electrical Parameters

Volts

Watts

Vars

Reactance, Transient, Sub-transient

Exciter Level

Mechanical Parameters

Rotating Mass

Throttle/Governor Characteristics

5.6 General Data Report Object

The GeneralDataReport object is a container object that can be used to exchange report data with control areas via TASE2 Block 8. This object is designed for sending one or more matrices of data. Each matrix may contain one or more rows of data with one or more user-defined column headings. Unlike the Transfer Account object defined in the IEC 60870-6-802 specification, there is no temporal relationship between rows.

This object allows the transfer of up to two different floating point, integer, or text value matrices in one object. The two matrices of each type can differ in the number of columns (matrices) of values. All data types, including the matrix data type, are standard data types defined in this part of IEC 60870, with the exception of a new text matrix data type and TransactionCode. Otherwise, only the abstract model and structure definitions differ from the standard.

5.6.1 Abstract Object Model

Object: **GeneralDataReport**

Attribute: GeneralDataReportReferenceNumber
Attribute: ReportName
Attribute: ReportDateAndTime
Attribute: TransactionCode
Attribute: NumberOfLocalReference
Attribute: ListOfLocalReference
Attribute: NumberOfFloatingPoint1 (NOTINCLUDED=0, INCLUDED not =0)
Constraint: NumberOfFloatingPoint1 not = 0 (INCLUDED)
  Attribute: NumberOfFloatingPoint1Rows
  Attribute: ListOfFloatingPoint1Values
Attribute: NumberOfFloatingPoint2 (NOTINCLUDED=0, INCLUDED not =0)
Constraint: NumberOfFloatingPoint2 not = 0 (INCLUDED)
  Attribute: NumberOfFloatingPoint2Rows
  Attribute: ListOfFloatingPoint2Values

Attribute: NumberOfInteger1 (NOTINCLUDED=0, INCLUDED not =0)
Constraint: NumberOfInteger1 not = 0 (INCLUDED)
  Attribute: NumberOfInteger1Rows
  Attribute: ListOfInteger1Values
Attribute: NumberOfInteger2 (NOTINCLUDED=0, INCLUDED not =0)
Constraint: NumberOfInteger2 not = 0 (INCLUDED)
  Attribute: NumberOfInteger2Rows
  Attribute: ListOfInteger2Values
Attribute: NumberOfText1 (NOTINCLUDED=0, INCLUDED not = 0)
Constraint: NumberOfText1 not = 0 (INCLUDED)
  Attribute: NumberOfText1Rows
  Attribute: ListOfText1Values
Attribute: NumberOfText2 (NOTINCLUDED=0, INCLUDED not =0)
Constraint: NumberOfText2 not = 0 (INCLUDED)
  Attribute: NumberOfText2Rows
  Attribute: ListOfText2Values

GeneralDataReportReferenceNumber
The GeneralDataReportReferenceNumber attribute specifies a unique reference value between the sender and the receiver to identify this particular report.

ReportName
The text identifier of this report.

ReportDateAndTime
The ReportDateAndTime attribute provides a means for the receiver to know when the sender sent this message. It is not related to the time to which the report data itself applies.

TransactionCode
The TransactionCode attribute provides status on the GeneralDataReport transaction. The values are shown below:

TransactionCode	Description
NEW	Original submittal of a newly proposed schedule or of a report
REQUESTED	Report or schedule is requested via the Account Request operation
REVISED	Revised version of previously sent schedule or report
CONFIRMED	Schedule or report has now been scheduled for actual implementation and is not subject to further approval cycles. A *curtailment, halt, revision,* or *deletion* after this stage requires a new schedule or report to be sent.
DELETED	Any time the originator of the schedule or report cancels the transaction (for example, due to equipment failure)

NumberOfLocal Reference
This attribute contains the count of entries in the ListOfLocalReference.

ListOfLocalReference
The ListOfLocalReference contains locally defined parameters that specify how the General Data Report Object is to be treated as a whole. The meaning of the parameter list is a local matter and is agreed upon among the participating parties for each type of General Data Report object, for all General Data Report objects, or for a combination of the two as needed.

NumberOfFloatingPoint1
The NumberOfFloatingPoint1 attribute is the count of FloatingPoint1 columns in this report. This count is also the flag to indicate whether FloatingPoint1 data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfFloatingPoint1Rows
The NumberOfFloatingPoint1Rows attribute is the number of entries (rows) in each FloatingPoint1 column in this report. The number of items in the FloatingPoint1 array is (NumberOfFloatingPoint1Rows × NumberOfFloatingPoint1).

ListOfFloatingPoint1Values
The ListOfFloatingPoint1Values attribute is the sequence of actual floating point values in the FloatingPoint1 array.

NumberOfFloatingPoint2
The NumberOfFloatingPoint2 attribute is the count of FloatingPoint2 columns in this report. This count is also the flag to indicate whether FloatingPoint2 data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfFloatingPoint2Rows
The NumberOfFloatingPoint2Rows attribute is the number of entries (rows) in each FloatingPoint2 column in this report. The number of items in the FloatingPoint2 array is (NumberOfFloatingPoint2Rows × NumberOfFloatingPoint2).

ListOfFloatingPoint2Values
The ListOfFloatingPoint2Values attribute is the sequence of actual floating point values in the FloatingPoint2 array.

NumberOfInteger1
The NumberOfInteger1 attribute is the count of Integer1 columns in this report. This count is also the flag to indicate whether Integer1data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfInteger1Rows
The NumberOfInteger1Rows attribute is the number of entries (rows) in each Integer1 column in this report. The number of items in the Integer1 array is (NumberOfInteger1Rows × NumberOfInteger1).

ListOfInteger1Values
The ListOfInteger1Values attribute is the sequence of actual integer values in the Integer1 array.

NumberOfInteger2
The NumberOfInteger2 attribute is the count of Integer2 columns in this report. This count is also the flag to indicate whether Integer2 data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfInteger2Rows
The NumberOfInteger2Rows attribute is the number of entries (rows) in each Integer2 column in this report. The number of items in the Integer2 array is (NumberOfInteger2Rows × NumberOfInteger2).

ListOfInteger2Values
The ListOfInteger2Values attribute is the sequence of actual integer values in the Integer2 array.

NumberOfText1
The NumberOfText1 attribute is the count of Text1 columns in this report. This count is also the flag to indicate whether Text1 data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfText1Rows
The NumberOfText1Rows attribute is the number of entries (rows) in each Text1 column in this report. The number of items in the Text1 array is (NumberOfText1Rows × NumberOfText1).

ListOfText1Values
The ListOfText1Values attribute is the sequence of actual text values in the Text1array.

NumberOfText2
The NumberOfText2 attribute is the count of Text2 columns in this report. This count is also the flag to indicate whether Text2 data is INCLUDED not = 0 or NOTINCLUDED=0 in this report.

NumberOfText2Rows
The NumberOfText2Rows attribute is the number of entries (rows) in each Text2 column in this report. The number of items in the Text2 array is (NumberOfText2Rows × NumberOfText2).

ListOfText2Values
The ListOfText2Values attribute is the sequence of actual text values in the Text2 array.

5.7 General Data Response Object

The GeneralDataResponse object is used to send a response to a Block 8 GeneralDataReport Object or any other Block 8 object that needs an acknowledgement. It further supports transactions involving multiple data transfers.

5.7.1 Abstract Object Model

Object: **GeneralDataResponse**

Attribute: ReportReferenceNumber
Attribute: ReportName
Attribute: ReportTimeStamp
Attribute: NumberOfLocalReference
Attribute: ListOfLocalReference
Attribute: ResponseData
Attribute: ResponseCode
Attribute: ResponseText

ReportReferenceNumber
The ReportReferenceNumber attribute specifies a unique reference value between the sender and receiver to identify this particular response. This number is the Reference Number of the received message.

ReportName
The ReportName (i.e. text identifier) from the received message. For example, for a response to a Generation Scheduling message this would be the Generation Unit/Schedule name from the received message Name attribute. This attribute further associates the response with the received message.

NumberOfLocal Reference
This attribute contains the count of entries in the ListOfLocalReference.

ListOfLocalReference
The ListOfLocalReference contains locally defined parameters that specify how the General Response Object is to be treated as a whole. The meaning of the parameter list is a local matter and is agreed upon among the participating parties for each type of General Data Report object, for all Transfer Account objects, or for a combination of the two as needed.

ReportTimeStamp
The ReportTimeStamp is the date and time stamp contained in the received message. This attribute associates the response with the received message.

ResponseData
The ResponseData is additional application supplied information that is used to associate the response with information from the received message.

ResponseCode
The ResponseCode is the numeric response code associated with the response. It has values which correspond to permitted responses to the TransactionCode defined in the GeneralDataReport object. The values are shown below:

ResponseCode	Description
RECEIVED	Acknowledgement that schedule or report has been successfully received
APPROVED	Acknowledgement that report or schedule has been approved
REJECTED	Acknowledgement that report or schedule has been rejected prior to CONFIRMATION
CURTAILED	Cancellation of CONFIRMED schedule prior to implementation by security authority to ensure reliability of electrical network or for other operational reasons
HALT	Cancellation of CONFIRMED schedule already implemented and in process by security authority to ensure reliability of electrical network or for other operational reasons

ResponseText
The ResponseText is application supplied readable text associated with the response.

6 MMS Types for Object Exchange

This clause defines the MMS Types to be used within TASE.2 for exchanging standard objects. The mapping of the objects onto these types is defined in clause 7. The MMS type definitions are defined in terms of ASN.1 value notation, following the MMS grammar for Data as defined in ISO 9506-1 and ISO 9506-2.

Throughout this clause, all field widths specified are maximum field widths. The process of ASN.1 encoding used within MMS may reduce the actual transmitted widths to the minimum required to represent the value being transmitted.

6.1 Supervisory Control and Data Acquisition Types

6.1.1 IndicationPoint Type Descriptions

The following foundation types are referenced in complex IndicationPoint Type Descriptions:

Data_Real floating-point: { format-width 32, exponent-width 8 }

Data_State bit-string:
{
State_hi[0],
State_lo[1],
Validity_hi[2],
Validity_lo[3],
CurrentSource_hi[4],
CurrentSource_lo[5],
NormalValue[6],
TimeStampQuality[7]
}

Data_Discrete integer {width 32 }

Data_Flags bit-string:
{
unused[0],
unused[1],
Validity_hi[2],
Validity_lo[3],
CurrentSource_hi[4],
CurrentSource_lo[5],
NormalValue[6],
TimeStampQuality[7]
}

Data_TimeStamp GMTBasedS

Data_TimeStampExtended **TimeStampExtended**

COV_Counter unsigned { width 16 }

The following complex types are used in transferring IndicationPoint object values:

Data_RealQ STRUCTURE
{
COMPONENT **Value** **Data_Real**,
COMPONENT **Flags** **Data_Flags**
}

Data_StateQ **Data_State**

Data_DiscreteQ STRUCTURE
{
COMPONENT **Value** **Data_Discrete**,
COMPONENT **Flags** **Data_Flags**
}

Data_RealQTimeTag STRUCTURE
{
COMPONENT **Value** **Data_Real**,
COMPONENT **TimeStamp** **Data_TimeStamp**,
COMPONENT **Flags** **Data_Flags**
}

```
Data_StateQTimeTag STRUCTURE
{
        COMPONENT TimeStamp     Data_TimeStamp,
        COMPONENT Flags         Data_State
}
Data_DiscreteQTimeTag STRUCTURE
{
        COMPONENT Value         Data_Discrete,
        COMPONENT TimeStamp     Data_TimeStamp,
        COMPONENT Flags         Data_Flags
}
Data_RealExtended STRUCTURE
{
        COMPONENT Value         Data_Real,
        COMPONENT TimeStamp     Data_TimeStamp,
        COMPONENT Flags         Data_Flags,
        COMPONENT COV           COVCounter
}
Data_StateExtended STRUCTURE
{
        COMPONENT TimeStamp     Data_TimeStamp,
        COMPONENT Flags         Data_State,
        COMPONENT COV           COVCounter
}
Data_DiscreteExtended STRUCTURE
{
        COMPONENT Value         Data_Discrete,
        COMPONENT TimeStamp     Data_TimeStamp,
        COMPONENT Flags         Data_Flags,
        COMPONENT COV           COVCounter
}
Data_RealQTimeTagExtended STRUCTURE
{
        COMPONENT Value         Data_Real,
        COMPONENT TimeStamp     Data_TimeStampExtended,
        COMPONENT Flags         Data_Flags
}

Data_StateQTimeTagExtended STRUCTURE
{
        COMPONENT TimeStamp     Data_TimeStampExtended,
        COMPONENT Flags         Data_State
}
Data_DiscreteQTimeTagExtended STRUCTURE
{
        COMPONENT Value         Data_Discrete,
        COMPONENT TimeStamp     Data_TimeStampExtended,
        COMPONENT Flags         Data_Flags
}
```

IndicationPointConfig STRUCTURE
{
COMPONENT **PointType** integer { width 8, range 0 .. 2 },
COMPONENT **QualityClass** integer { width 8, range 0 .. 1 },
COMPONENT **NormalSource** integer { width 8, range 0 .. 3 },
COMPONENT **TimeStampClass** integer { width 8, range 0 .. 1 },
COMPONENT **COVClass** integer { width 8, range 0 .. 1 }
}

6.1.2 ControlPoint Type Descriptions

The following foundation types are referenced in complex type descriptions:

Control_Command integer { width 16 }

Control_Setpoint_Real floating-point { format-width 32, exponent-width 8 }

Control_Setpoint_Discrete integer { width 16 }

SBO_CheckBackName integer { width 16 }

SelectState boolean

TagFlags bit-string:
{
tag_hi [0],
tag_lo [1],
tag_state[2]
}

TextString VisibleString { width 255 }

The following complex type descriptions are used in accessing ControlPoint object values:

SBO STRUCTURE
{
COMPONENT **TimeOut** **Data_TimeStamp**,
COMPONENT **Select** **SelectState**
}
Tag_Value STRUCTURE
{
COMPONENT **Flags** **TagFlags**,
COMPONENT **Reason** **TextString**
}
ControlConfig STRUCTURE
{
COMPONENT **ControlPointType** integer { width 8, range 0..2 },
COMPONENT **SetPointType** integer { width 8, range 0 .. 2 },
COMPONENT **DeviceClass** integer { width 8, range 0 .. 1 },
COMPONENT **TagClass** integer { width 8, range 0 .. 1 }
}

6.1.3 Protection Equipment Type Descriptions

The following foundation types are used to build complex types for report protection equipment events.

SingleFlags bit-string:

```
{
    ElapsedTimeValidity[0],
    Blocked[1],
    Substituted[2],
    Topical[3],
    EventValidity[4],
    unused[5],
    EventState_hi[6],
    EventState_lo[7]
}
```

EventFlags bit-string:

```
{
    General[0],
    Phase1[1],
    Phase2[2],
    Phase3[3],
    Earth[4],
    Reverse[5],
    unused[6],
    unused[7]
}
```

PackedFlags bit-string:

```
{
    ElapsedTimeValidity[0],
    Blocked[1],
    Substituted[2],
    Topical[3],
    EventValidity[4],
    unused[5],
    unused[6],
    unused[7]
}
```

The following complex types are used to report protection equipment events.

SingleProtectionEvent STRUCTURE

```
{
    COMPONENT SingleEventFlags      SingleFlags,
    COMPONENT OperatingTime         TimeIntervalL16,
    COMPONENT EventTime             TimeStampExtended
}
```

PackedProtectionEvent STRUCTURE

```
{
    COMPONENT PackedEvent           EventFlags,
    COMPONENT PackedEventFlags      PackedFlags,
    COMPONENT OperatingTime         TimeIntervalL16,
    COMPONENT EventTime             TimeStampExtended
}
```

6.2 Transfer Account Types

The following foundation types are referenced in complex type descriptions:

CommodityClass	integer { width 16 }
Data_Discrete	integer { width 32 }
Data_Real	floating point { format-width 32, exponent-width 8 }
Data_TimeStamp	**GMTBasedS**
Duration	**TimeIntervalS**
IntegerId	integer { width 32 }
InterchangeId	integer { width 32 }
NameString	visiblestring { width 32 characters, varying }
NumIntegers	integer { width 16 }
NumFloats	integer { width 16 }
NumPeriods	integer { width 16 }
NumProfiles	integer { width 16 }
NumSegs	integer { width 16 }
Price	floating point { format-width 32, exponent-width 8 }
ReferenceNum	integer { width 32 }
ScheduleTime	**GMTBasedS**
TransactionAmount	floating point { format-width 32, exponent-width 8 }
TransactionCode	Integer { width 32 }
TransmissionSegCode	integer { width 16 }
UtilityId	integer { width 32 }

6.2.1 TransferAccount Type Descriptions

TASegmentsPeriodic STRUCTURE
{

COMPONENTS	**TransferAccountRef**	**ReferenceNum,**
COMPONENTS	**SendUtility**	**UtilityId,**
COMPONENTS	**RecvUtility**	**UtilityId,**
COMPONENTS	**SellingUtility**	**UtilityId,**
COMPONENTS	**BuyingUtility**	**UtilityId,**
COMPONENTS	**TimeStamp**	**Data_TimeStamp,**
COMPONENTS	**TransactionCode**	**TransactionCode,**
COMPONENTS	**Name**	**NameString,**
COMPONENTS	**StartTime**	**ScheduleTime,**
COMPONENTS	**PeriodResolution**	**TimeIntervalS,**
COMPONENTS	**NumberLocalReferences**	**NumIntegers,**
COMPONENTS	**NumberSegments**	**NumSegs,**
COMPONENTS	**NumberFloatIds**	**NumFloats,**
COMPONENTS	**NumberIntegerIds**	**NumIntegers,**
COMPONENTS	**NumberPeriods**	**NumPeriods**

}

TANoSegmentsPeriodic STRUCTURE
{

COMPONENTS	**TransferAccountRef**	**ReferenceNum,**
COMPONENTS	**SendUtility**	**UtilityId,**
COMPONENTS	**RecvUtility**	**UtilityId,**
COMPONENTS	**SellingUtility**	**UtilityId,**
COMPONENTS	**BuyingUtility**	**UtilityId,**
COMPONENTS	**TimeStamp**	**Data_TimeStamp,**
COMPONENTS	**TransactionCode**	**TransactionCode,**
COMPONENTS	**Name**	**NameString,**

```
        COMPONENTS      StartTime               ScheduleTime,
        COMPONENTS      PeriodResolution        TimeIntervalS,
        COMPONENTS      NumberLocalReferences   NumIntegers,
        COMPONENTS      NumberFloatIds          NumFloats,
        COMPONENTS      NumberIntegerIds        NumIntegers,
        COMPONENTS      NumberPeriods           NumPeriods
}

TASegmentsProfile STRUCTURE
{
        COMPONENTS      TransferAccountRef      ReferenceNum,
        COMPONENTS      SendUtility             UtilityId,
        COMPONENTS      RecvUtility             UtilityId,
        COMPONENTS      SellingUtility          UtilityId,
        COMPONENTS      BuyingUtility           UtilityId,
        COMPONENTS      TimeStamp               Data_TimeStamp,
        COMPONENTS      TransactionCode         TransactionCode,
        COMPONENTS      Name                    NameString,
        COMPONENTS      NumberLocalReferences   NumIntegers,
        COMPONENTS      NumberSegments          NumSegs,
        COMPONENTS      NumberProfileValues     NumProfiles
}

TANoSegmentsProfile STRUCTURE
{
        COMPONENTS      TransferAccountRef      ReferenceNum,
        COMPONENTS      SendUtility             UtilityId,
        COMPONENTS      RecvUtility             UtilityId,
        COMPONENTS      SellingUtility          UtilityId,
        COMPONENTS      BuyingUtility           UtilityId,
        COMPONENTS      TimeStamp               Data_TimeStamp,
        COMPONENTS      TransactionCode         TransactionCode,
        COMPONENTS      Name                    NameString,
        COMPONENTS      NumberLocalReferences   NumIntegers,
        COMPONENTS      NumberProfileValues     NumProfiles
}
```

6.2.2 TransmissionSegment Type Descriptions

```
TATransmissionSegment STRUCTURE
{
        COMPONENT TransmissionReference         ReferenceNum,
        COMPONENT UtilWheeling                  UtilityId,
        COMPONENT UtilPaying                    UtilityId,
        COMPONENT TransmissionSegType           TransmissionSegCode,
        COMPONENT UtilIn                        UtilityId,
        COMPONENT UtilOut                       UtilityId,
        COMPONENT InterchangePtIn               InterchangeId,
        COMPONENT InterchangePtOut              InterchangeId,
        COMPONENT InterchangePt                 InterchangeId,
        COMPONENT NumberFloatIds                NumFloats,
        COMPONENT NumberIntegerIds              NumIntegers
}
```

6.2.3 Transmission Segment Type Descriptions

TATransmissionSegmentProfile STRUCTURE
{

COMPONENT **TransmissionReference**	**ReferenceNum,**
COMPONENT **UtilWheeling**	**UtilityId,**
COMPONENT **UtilPaying**	**UtilityId,**
COMPONENT **TransmissionSegType**	**TransmissionSegCode,**
COMPONENT **UtilIn**	**UtilityId,**
COMPONENT **UtilOut**	**UtilityId,**
COMPONENT **InterchangePtIn**	**InterchangeId,**
COMPONENT **InterchangePtOut**	**InterchangeId,**
COMPONENT **InterchangePt**	**InterchangeId,**
COMPONENT **NumberProfileValues**	**NumProfiles**

}

6.2.4 ProfileValue Type Descriptions

TAProfileValue STRUCTURE
{

COMPONENT **RampStartTime**	**ScheduleTime,**
COMPONENT **RampDuration**	**Duration,**
COMPONENT **ProfilePrice**	**Price,**
COMPONENT **ProfileTargetClass**	**CommodityClass,**
COMPONENT **ProfileTarget**	**TransactionAmount**

}

6.2.5 Account Request Type Descriptions

AccountRequest STRUCTURE
{

COMPONENT **ReferenceTar**	**ReferenceNum,**
COMPONENT **StartTime**	**ScheduleTime,**
COMPONENT **Duration**	**Duration,**
COMPONENT **RequestId**	**ReferenceNum,**
COMPONENT **TaConditionsRequested**	**TAConditions**

}

6.3 Device Outage Type Descriptions

The following foundation types are referenced in complex type descriptions for the DeviceOutage object:

ActionId	integer {width 32}
ClassId	integer {width 16}
CommentString	VisibleString 128
Data_Real	floating-point {format-width 32, exponent-width 8}
DeviceId	integer {width 32}
DeviceName	visiblestring {width 32 characters, varying}
DeviceNumber	integer {width 32}
Number	integer {width 16}
OutageAmountTypeId	integer {width 16}
OutagePeriodId	integer {width 16}
OutageTypeId	integer {width 16}
PlanTypeId	integer {width 16}
ReferenceNum	integer {width 32}

ScheduleTime	GMTBasedS
StationNameString	VisibleString 32
TimeStampS	GMTBasedS
UtilityId	integer {width 32}

DeviceOutage composite type definitions are:

DONewRevSched STRUCTURE
{
COMPONENT **OutageReferenceId** **ReferenceNum**,
COMPONENT **OwningUtilityId** **UtilityId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **StationName** **StationNameString**,
COMPONENT **DeviceName** **DeviceName**,
COMPONENT **DeviceType** **DeviceId**,
COMPONENT **DeviceNumber** **Number**,
COMPONENT **DeviceRating** **Data_Real**,
COMPONENT **ActivityDateAndTime** **ScheduleTime**,
COMPONENT **PlanType** **PlanTypeId**,
COMPONENT **PlannedOpenOrOutOfServiceDateAndTime** **ScheduleTime**,
COMPONENT **PlannedCloseOrInServiceDateAndTime** **ScheduleTime**,
COMPONENT **OutagePeriod** **OutagePeriodId**,
COMPONENT **OutageType** **OutageTypeId**,
COMPONENT **OutageTypeAmount** **OutageTypeAmountId**,
COMPONENT **Amount** **Data_Real**,
COMPONENT **UpperOperatingLimit** **Data_Real**,
COMPONENT **LowerOperatingLimit** **Data_Real**,
COMPONENT **Class** **ClassId**,
COMPONENT **Comments** **CommentString**,
COMPONENT **OutageEffect** **CommentString**
}

DOCancel STRUCTURE
{
COMPONENT **OutageReferenceId** **ReferenceNum**,
COMPONENT **OwningUtilityId** **UtilityId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **StationName** **StationNameString**,
COMPONENT **DeviceName** **DeviceName**,
COMPONENT **DeviceType** **DeviceId**,
COMPONENT **DeviceNumber** **Number**,
COMPONENT **DeviceRating** **Data_Real**,
COMPONENT **ActivityDateAndTime** **ScheduleTime**,
COMPONENT **Comments** **CommentString**,
COMPONENT **OutageEffect** **CommentString**
}

DOActual STRUCTURE
{
COMPONENT **OutageReferenceId** **ReferenceNum**,
COMPONENT **OwningUtilityId** **UtilityId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **StationName** **StationNameString**,

COMPONENT **DeviceName**	**DeviceName**,
COMPONENT **DeviceType**	**DeviceId**,
COMPONENT **DeviceNumber**	**Number**,
COMPONENT **DeviceRating**	**Data_Real**,
COMPONENT **ActivityDateAndTime**	**ScheduleTime**,
COMPONENT **Action**	**ActionId**,
COMPONENT **AffectedAmount**	**Data_Real**,
COMPONENT **Comments**	**CommentString**,
COMPONENT **OutageEffect**	**CommentString**

}

6.4 InformationBuffer Type Descriptions

The following foundation types are referenced in complex type descriptions for the InformationBuffer object:

ReferenceNum	integer {width 32}
Number	integer {width 32}

The following types are used in exchanging the Information Message object types:

InfoMessHeader STRUCTURE
{

COMPONENT **InfoReference**	**ReferenceNum,**
COMPONENT **LocalReference**	**ReferenceNum,**
COMPONENT **MessageId**	**ReferenceNum,**
COMPONENT **Size**	**Number**

}

InfoBufXX OCTET STRING {width XX octets}

where XX is the number of octets in the buffer. Any number of bytes are permitted, although the maximum buffer size should not exceed the maximum MMS PDU size.

Examples: 64 byte buffer = InfoBuf64, 256 byte buffer = InfoBuf256, 1024 byte buffer = InfoBuf1024. Note that leading zeros are not permitted.

6.5 Power Plant Type Descriptions

The following foundation types are used throughout the Power Plant - EMS Data Exchange type definitions:

AvailReasonCode	integer {width 32}
Capacity	floating-point {format-width 32, exponent-width 8}
DateAndTime	GMTBasedS
LFCReasonCode	integer {width 32}
PlantId	integer {width 32}
Price	floating-point {format-width 32, exponent-width 8}
RampRate	floating-point {format-width 32, exponent-width 8}
ReferenceNum	integer {width 32}
ReportStatus	integer {width 32}
UnitId	integer {width 32}

```
UnitCapacity STRUCTURE
{
        COMPONENT CapacityClass          bitstring { gross[0], net[1] },
        COMPONENT GrossMaxCapacity       Capacity,
        COMPONENT GrossMinCapacity       Capacity,
        COMPONENT NetMaxCapacity         Capacity,
        COMPONENT NetMinCapacity         Capacity
}

RampRates STRUCTURE
{
        COMPONENT MaxRampRateUp          RampRate,
        COMPONENT MaxRampRateDown        RampRate
}

AvailabilityClass STRUCTURE
{
        COMPONENT AvailFlags             bitstring
                                         {
                                             online      [0],
                                             LFC         [1],
                                             dispatch    [2],
                                             regulating  [3],
                                             manual      [4],
                                             reserve     [5],
                                             blocked_hi  [6],
                                             blocked_lo  [7]
                                         }
        COMPONENT TimeToOnline                   GMTBasedS
        COMPONENT ReasonForNoLFC                 LFCReasonCode
}
```

6.5.1 Availability Report Type Descriptions

```
Available STRUCTURE
{
        COMPONENT AvailabilityReferenceId   ReferenceNum,
        COMPONENT PlantReferenceId          PlantId,
        COMPONENT UnitId                    UnitId,
        COMPONENT TimeStamp                 TimeStampS,
        COMPONENT StartDateAndTime          DateAndTime,
        COMPONENT EndDateAndTime            DateAndTime,
        COMPONENT Duration                  TimeIntervalM,
        COMPONENT ProvidingReserve          boolean,
        COMPONENT ReportStatus              ReportStatus,
        COMPONENT Impact
        COMPONENT PriceImpact               Price,
        COMPONENT RampImpact                RampRates,
        COMPONENT CapacityImpact            UnitCapacity,
        COMPONENT Availability              AvailabilityClass,
        COMPONENT Comment                   VisibleString { width 255 }
}
```

UnAvailable STRUCTURE
{
COMPONENT **AvailabilityReferenceId** **ReferenceNum**,
COMPONENT **PlantReferenceId** **PlantId**,
COMPONENT **UnitId** **UnitId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **StartDateAndTime** **DateAndTime**,
COMPONENT **EndDateAndTime** **DateAndTime**,
COMPONENT **Duration** TimeIntervalM,
COMPONENT **ProvidingReserve** boolean,
COMPONENT **ReportStatus** **ReportStatus**,
COMPONENT **ReasonForUnavailable** **AvailReasonCode**,
COMPONENT **Comment** VisibleString { width 255 }
}

6.5.2 Real Time Status Type Descriptions

StatusAvailable STRUCTURE
{
COMPONENT **AvailabilityReferenceId** **ReferenceNum**,
COMPONENT **PlantReferenceId** **PlantId**,
COMPONENT **UnitId** **UnitId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **RampStatus** **RampRates**,
COMPONENT **CapacityStatus** **UnitCapacity,**
COMPONENT **AvailabilityStatus** **AvailabilityClass**,
COMPONENT **ProvidingReserve** boolean
}

StatusUnAvailable STRUCTURE
{
COMPONENT **AvailabilityReferenceId** **ReferenceNum**,
COMPONENT **PlantReferenceId** **PlantId**,
COMPONENT **UnitId** **UnitId**,
COMPONENT **TimeStamp** **TimeStampS**,
COMPONENT **ReasonForUnavailable** **AvailReasonCode**,
COMPONENT **ProvidingReserve** boolean
}

6.5.3 Forecast Type Descriptions

Forecast STRUCTURE
{
COMPONENT **ForecastScheduleReferenceId** **ReferenceNum**,
COMPONENT **PlantReferenceId** **PlantId**,
COMPONENT **UnitId** **UnitId**,
COMPONENT **StartTime** **GMTBasedS**,
COMPONENT **PeriodResolution** **TimeIntervalS**,
COMPONENT **NumPeriods** integer,
COMPONENT **ForecastType** bitstring { generation[0], reserve[1] }
}

6.5.4 Curve Type Descriptions

CurveLimit floating-point { format-width 32, exponent-width 8 }

CurveType bit-string:

{
HEAT_RATE [0]
IO [1]
IHR [2]
MVAR_CAP [3]
COST [4]
OPACITY [5]
SOX [6]
NOX [7]
CO2 [8]
Unused [9-15]
User defined [16-31]
}

Coefficient floating-point { format-width 32, exponent-width 8 }

CurveSegmentDescription STRUCTURE
{
COMPONENT **Order** integer { width 16 },
COMPONENT **LowRange** **CurveLimit,**
COMPONENT **HighRange** **CurveLimit,**
COMPONENT **NumberOfSegments** **NumSegs**
}

Curve STRUCTURE
{
COMPONENT **CurveName** visiblestring {width 32 characters, varying}
COMPONENT **PlantReferenceId** **PlantId,**
COMPONENT **UnitId** **UnitId,**
COMPONENT **CurveType** **CurveType,**
COMPONENT **NumberOfSegments** **NumSegs**
}

6.6 Power System Dynamics

No new data types are required to support Power System Dynamics. These are mapped to the variables of simple MMS base types.

6.7 Matrix Data Types

The Matrix data types are used to represent sequences or lists of logical records, in which each logical record contains a set of values to be reported for that step in the sequence. An example of such a sequence is an interchange schedule, in which the scheduled quantities (energy, capacity, etc.) and their associated costs are reported for each of a given set of time intervals. The Matrix types allow for arbitrary sets of values and arbitrarily long sequences to be represented, even though the particular mix of data types, semantics, and length of sequences may not be known until run time.

The basic representation of a sequence of homogeneous logical records within an MMS Information Report using the Matrix Data Types uses the following sequence of MMS Named Variables:

a) The report must contain a *header* variable, which is generally specific to the object model which contains the sequence or list. The header variable must include the expected length of the sequence or list, as well as a count of the total number of integer values and the total number of floating point values which are to be reported for each record in the sequence. The number and identity of the integer and floating point values may change from report to report, but will be consistent for each logical record within a given report.

b) If there are floating point values to be included in the logical records:

The report will contain a variable with name **Matrix_Id** and type **MatrixId** for each floating point value which is to be included in the logical records. The number of **Matrix_Id** variables for floating point values expected can be determined from the values in the header variable.

For each logical record being reported, a variable of type **FloatArrayXX** will be reported which contains the floating point values for the record. The size of **FloatArrayXX** (XX is the array dimension) is arbitrary, but must be greater than or equal to the number of floating point values to be included. If the array is larger than the number of floating point values included in the logical records, the values are filled from the low order (starting from FloatArrayXX[0]) and the remainder are ignored.

c) If there are integer values to be included in the logical records:

The report will contain a variable with name **Matrix_Id** and type **MatrixId** for each integer value which is to be included in the logical records. The number of **Matrix_Id** variables for integer values expected can be determined from the values in the header variable.

For each logical record being reported, a variable of type **IntegerArrayXX** will be reported which contains the integer values for the record. The size of **IntegerArrayXX** (XX is the array dimension) is arbitrary, but must be greater than or equal to the number of integer values to be included. If the array is larger than the number of integer values included in the logical records, the values are filled from the low order (starting from IntegerArrayXX[0]) and the remainder are ignored.

d) If there are text values to be included in the logical records:

The report will contain a variable with name **Matrix_Id** and type **MatrixId** for each text value which is to be included in the logical records. The number of **Matrix_Id** variables for text values expected can be determined from the values in the header variable.

For each logical record being reported, a variable of type **Text32ArrayXX** will be reported which contains the text values for the record. The size of **Text32ArrayXX** (XX is the array dimension) is arbitrary, but must be greater than or equal to the number of text values to be included. If the array is larger than the number of text values included in the logical records, the values are filled from the low order (starting from Text32ArrayXX[0]) and the remainder are ignored.

Each **FloatArrayXX, IntegerArrayXX, and Text32ArrayXX** tuple constitutes a logical record (and hence a row of the Matrix).

The following types are defined :

MatrixId	integer { width 32 }
IntegerArrayXX	ARRAY[XX] of integer { width 32 }
FloatArrayXX	ARRAY[XX] of floating point { format-width 32, exponent-width 8 }
Text32ArrayXX	ARRAY{XX} of VisibleString {width32}

where XX is an arbitrary array dimension.

Example: 256 byte array = IntegerArray256, 1024 byte array = IntegerArray1024

6.8 GeneralDataReport Type Descriptions

Most of the foundation types referenced in the complex GeneralDataReport type descriptions below are the same as those used in Transfer Accounts and are defined in the Transfer Account Types section. The following new foundation types are also referenced in the complex GeneralDataReport type description:

TransactionCode Integer { width 32 }
NumRows Integer { width 16 }

The following complex type is used for exchanging the General Data Reports:

GeneralDataReport STRUCTURE
{

COMPONENT	**GeneralDataReportReferenceNumber**	**ReferenceNum**
COMPONENT	**ReportName**	**NameString**
COMPONENT	**ReportDateAndTime**	**TimeStampS**
COMPONENT	**TransactionCode**	**TransactionCode**
COMPONENT	**NumberOfLocalReference**	**NumIntegers**
COMPONENT	**NumberOfFloatingPoint1**	**NumFloats**
COMPONENT	**NumberOfFloatingPoint1Rows**	**NumRows**
COMPONENT	**NumberOfFloatingPoint2**	**NumFloats**
COMPONENT	**NumberOfFloatingPoint2Rows**	**NumRows**
COMPONENT	**NumberOfInteger1**	**NumIntegers**
COMPONENT	**NumberOfInteger1Rows**	**NumRows**
COMPONENT	**NumberOfInteger2**	**NumIntegers**
COMPONENT	**NumberOfInteger2Rows**	**NumRows**
COMPONENT	**NumberOfText1**	**NumIntegers**
COMPONENT	**NumberOfText1Rows**	**NumRows**
COMPONENT	**NumberOfText2**	**NumIntegers**
COMPONENT	**NumberOfText2Rows**	**NumRows**

}

6.9 GeneralDataResponse Type Descriptions

The following complex type is used for exchanging the General Data Reports:

GeneralDataResponse STRUCTURE
{

COMPONENT	**ReportReferenceNumber**	**ReferenceNumber**
COMPONENT	**ReportName**	**NameString**
COMPONENT	**ReportTimeStamp**	**Data_TimeStamp**
COMPONENT	**NumberOfLocalReferences**	**NumIntegers**
COMPONENT	**ResponseData**	Integer {width 32}
COMPONENT	**ResponseCode**	**TransactionCode**
COMPONENT	**ResponseText**	**CommentString**

}

7 Mapping of Object Models to MMS Types

7.1 Supervisory Control and Data Mapping

7.1.1 Indication Object Mapping

This clause defines the mapping of each object attributes onto MMS. In general, most objects are represented by one or more MMS Named Variables of the predefined TASE.2 types from clause 6.

PointName
Maps to an MMS variable identifier (either VMD specific or Domain specific)

PointType
Used in selecting the named type of the variable. If COVClass is NOCOV, the type of the MMS variable is selected according to the following criteria:

PointType	QualityClass	TimeStampClass	Map to type:
REAL	NOQUALITY	NOTIMESTAMP	**Data_Real**
STATE	NOQUALITY	NOTIMESTAMP	**Data_State**
DISCRETE	NOQUALITY	NOTIMESTAMP	**Data_Discrete**
REAL	QUALITY	NOTIMESTAMP	**Data_RealQ**
STATE	QUALITY	NOTIMESTAMP	**Data_StateQ**
DISCRETE	QUALITY	NOTIMESTAMP	**Data_DiscreteQ**
REAL	QUALITY	TIMESTAMP	**Data_RealQTimeTag**
STATE	QUALITY	TIMESTAMP	**Data_StateQTimeTag**
DISCRETE	QUALITY	TIMESTAMP	**Data_DiscreteQTimeTag**
REAL	QUALITY	TIMESTAMPEXTENDED	**Data_RealQTimeTagExtended**
STATE	QUALITY	TIMESTAMPEXTENDED	**Data_StateQTimeTagExtended**
DISCRETE	QUALITY	TIMESTAMPEXTENDED	**Data_DiscreteQTimeTagExtended**

If COVClass is COV, the following criteria are used:

PointType	Map to type:
REAL	**Data_RealExtended**
STATE	**Data_StateExtended**
DISCRETE	**Data_DiscreteExtended**

The PointType attribute may optionally be mapped to the **PointType** component of an MMS named variable of type **IndicationPointConfig** with the following interpretation: 0=STATE, 1=DISCRETE, 2=REAL.

PointRealValue
If present, maps to either the value of an MMS variable of type **Data_Real** (if QualityClass and TimeStampClass are NOQUALITY, NOTIMESTAMP) or to the **Value** COMPONENT of the MMS variable.

PointStateValue
If present, maps to either the value of an MMS variable of type **Data_State** (if QualityClass and TimeStampClass are NOQUALITY, NOTIMESTAMP) or to bits **State_hi** and **State_lo** of the **Flags** COMPONENT of the MMS variable.

PointDiscreteValue
If present, maps to either the value of the MMS variable of type **Data_Discrete** (if QualityClass and TimeStampClass are NOQUALITY, NOTIMESTAMP) or to the **Value** COMPONENT of the MMS variable.

QualityClass
Used in selecting the named type of the variable (see above). The QualityClass attribute may also be optionally mapped to the **QualityClass** component of an MMS Named Variable of type **IndicationPointConfig** with the following interpretation: NOQUALITY=0, QUALITY=1.

Validity
If present, maps to bits 2 and 3 (**Validity_hi**, **Validity_lo**) of the **Flags** COMPONENT with the following values: VALID = 0, HELD=1, SUSPECT=2, NOTVALID=3.

CurrentSource
If present, maps to bits 4 and 5 (**CurrentSource_hi**, **CurrentSource_lo**) of the **Flags** COMPONENT with the following values: TELEMETERED=0, CALCULATED=1, ENTERED=2, ESTIMATED=3.

NormalSource
The NormalSource attribute may be optionally mapped to the **NormalSource** component of an MMS Named Variable of type **IndicationPointConfig** with the following interpretation: TELEMETERED=0, CALCULATED=1, ENTERED=2, ESTIMATED=3.

NormalValue
If present, maps to bit 6 (**NormalValue**) of the **Flags** COMPONENT with the following values: NORMAL=0, ABNORMAL=1.

TimeStampClass
Used in selecting the named type of the variable (see above). The TimeStampClass attribute may also be optionally mapped to the **TimeStampClass** component of an MMS Named Variable of type **IndicationPointConfig** with the following interpretation: NOTIMESTAMP=0, TIMESTAMP=1, TIMESTAMPEXTENDED=2.

TimeStamp
If present, maps to the **TimeStamp** COMPONENT.

TimeStampQuality
If present, maps to bit 7 (**TimeStampQuality**) of the **Flags** COMPONENT with the following values: VALID=0, INVALID=1.

COVClass
Used in selecting the named type of the variable (see above). The COVClass attribute may also be optionally mapped to the **COVClass** component of an MMS Named Variable of type **IndicationPointConfig** with the following interpretation: NOCOV=0, COV=1.

COVCounter
If present, maps to an MMS variable of type **COV_Counter**.

7.1.2 ControlPoint Object Mapping

ControlPointName
Maps to an MMS variable identifier (either VMD specific or Domain specific).

ControlPointType
Used in selecting the named type of the variable. The type of the MMS variable is selected according to the following criteria (all other combinations are invalid):

ControlPointType	SetPointType	Map to type:
COMMAND	Not applicable	**Control_Command**
SETPOINT	REAL	**Control_Setpoint_Real**
SETPOINT	DISCRETE	**Control_Setpoint_Discrete**

The ControlPointType attribute may also be optionally mapped to the **ControlPointType** component of an MMS Named Variable of type **ControlConfig** with the following interpretation: COMMAND=0, SETPOINT=1.

CommandValue
Maps to the value of an MMS variable of type **Control_Command**.

SetPointType
Used in selecting the named type of the variable (see above). The Setpoint attribute may also be optionally mapped to the **SetPointType** component of an MMS Named Variable of type **ControlConfig** with the following interpretation: 1=DISCRETE, 2=REAL.

SetpointRealValue
If present, maps to the value of an MMS variable of type **Control_Setpoint_Real**.

SetpointDiscreteValue
If present, maps to the value of an MMS variable of type **Control_Setpoint_Discrete**.

DeviceClass
The DeviceClass attribute may be optionally mapped to the **DeviceClass** component of an MMS Named Variable of type **ControlConfig** with the following interpretation: 0=NONSBO, 1=SBO.

CheckBackName
If present, maps to an MMS variable of type **SBO_CheckBackName**.

State
If present, maps to the **Select** COMPONENT of an MMS variable of type **SBO** with the following interpretation: SELECTED=TRUE, NOTSELECTED=FALSE.

Timeout
If present, maps to the **TimeOut** COMPONENT of an MMS variable of type **SBO**.

TagClass
The TagClass attribute may be optionally mapped to the **TagClass** component of an MMS Named Variable of type **ControlConfig** with the following interpretation: 1=DISCRETE, 2=REAL.

Tag
If present, maps to bits 0 and 1 (**Tag_hi** and **Tag_lo**) of the **Flags** COMPONENT of an MMS variable of type **Tag_Value** with the following interpretation: NO-TAG=0, OPEN- AND-CLOSE-INHIBIT=1, CLOSE-ONLY-INHIBIT=2.

State
If present, maps to bit 2 (**Tag_state**) of the **Flags** COMPONENT of an MMS variable of type **Tag_Value** with the following interpretation: IDLE=0, ARMED=1

Reason
If present, maps to the **Reason** COMPONENT of an MMS variable of type **Tag_Value**.

7.1.3 Protection Event Mapping

This subclause defines the mapping of each attribute of the protection event model to MMS types. In general, protection events are mapped onto MMS variables. The recommended method of generating protection event reporting is to include the MMS variables in a TASE.2 Data Set and have the Data Set reported using a Data Set Transfer Set with DSTransmissionPars RBE True and DSConditions including ObjectChange True.

Name
The Name attribute is mapped to an MMS Named variable identifier. Note that the name must uniquely identify both the protection device and the protection event type.

ElapsedTimeValidity
The ElapsedTimeValidity attribute is mapped onto bit 0 (**ElapsedTimeValidity**) of the **SingleFlags** or **PackedFlags** COMPONENT of the MMS variable representing the protection event, with the value 0 representing VALID and 1 representing INVALID.

Blocked
The Blocked attribute is mapped onto bit 1 (**Blocked**) of the **SingleFlags** or **PackedFlags** COMPONENT of the MMS variable representing the protection event, with the value 0 representing NOTBLOCKED and 1 representing BLOCKED.

Substituted
The Substituted attribute is mapped onto bit 2 (**Substituted**) of the **SingleFlags** or **PackedFlags** COMPONENT of the MMS variable representing the protection event, with the value 0 representing NOTSUBSTITUTED and 1 representing SUBSTITUTED.

Topical
The Topical attribute is mapped onto bit 3 (**Topical**) of the **SingleFlags** or **PackedFlags** COMPONENT of the MMS variable representing the protection event, with the value 0 representing TOPICAL and 1 representing NOTTOPICAL.

EventValidity
The EventValidity attribute is mapped onto bit 4 (**EventValidity**) of the **SingleFlags** or **PackedFlags** COMPONENT of the MMS variable representing the protection event, with the value 0 representing VALID and 1 representing INVALID.

ProtectionClass
The ProtectionClass attribute is used to select the type of MMS variable to represent the event. The value of SINGLE implies the use of a variable of type **SingleProtectionEvent**, and the value of PACKED implies the use of a variable of type **PackedProtectionEvent**.

EventState
The EventState attribute is mapped to bits 6 and 7 (**EventState_hi** and **EventState_lo**) of the **SingleFlags** COMPONENT of an MMS variable of type **SingleProtectionEvent**, with 1 denoting OFF, 2 denoting ON and 0 or 3 denoting INVALID.

EventDuration
The EventDuration attribute maps to the **OperatingTime** COMPONENT of an MMS variable of type **SingleProtectionEvent**.

EventTime
The EventTime attribute maps to the **EventTime** COMPONENT of an MMS variable of type **SingleProtectionEvent**.

EventClass
The EventClass attribute is not mapped. The value of this attribute is implied by the identifier of the MMS variable used to represent the packed protection event, and must be known and agreed to by the involved parties.

StartGeneral
The StartGeneral attribute is mapped to bit 0 (**General**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

StartPhase1
The StartPhase1 attribute is mapped to bit 1 (**Phase1**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

StartPhase2
The StartPhase2 attribute is mapped to bit 2 (**Phase2**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

StartPhase3
The StartPhase3 attribute is mapped to bit 3 (**Phase3**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

StartEarth
The StartEarth attribute is mapped to bit 4 (**Earth**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

StartReverse
The StartReverse attribute is mapped to bit 5 (**Reverse**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOSTART and 1 representing START.

DurationTime
The DurationTime attribute is mapped to the **OperatingTime** COMPONENT of an MMS variable of type **PackedProtectionEvent**.

StartTime
The StartTime attribute is mapped to the **EventTime** COMPONENT of an MMS variable of type **PackedProtectionEvent**.

TripGeneral
The TripGeneral attribute is mapped to bit 0 (**General**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOTRIP and 1 representing TRIP.

TripPhase1
The TripPhase1 attribute is mapped to bit 1 (**Phase1**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOTRIP and 1 representing TRIP.

TripPhase2
The TripPhase2 attribute is mapped to bit 2 (**Phase2**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOTRIP and 1 representing TRIP.

TripPhase3
The TripPhase3 attribute is mapped to bit 3 (**Phase3**) of the **EventFlags** COMPONENT of an MMS variable of type **PackedProtectionEvent**, with 0 representing NOTRIP and 1 representing TRIP.

OperatingTime
The OperatingTime attribute is mapped to the **OperatingTime** COMPONENT of an MMS variable of type **PackedProtectionEvent**.

TripTime
The TripTime attribute is mapped to the **EventTime** COMPONENT of an MMS variable of type **PackedProtectionEvent**.

7.2 Transfer Accounts Mapping

7.2.1 TransferAccount Mapping

Transfer Accounts are only visible through the Transfer Account Reporting and Account Query mechanisms as defined in IEC 60870-6-503. The Transfer Accounts are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the Transfer Account object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

Each of the classes of Transfer Accounts contain header information and optionally one or more associated lists of data (ListOfLocalReference, ListOfTransmissionSegment, ListOfProfileValues, and ListOfPeriodicValues). The lists of data are represented as follows:

ListOfLocalReference – sequence of MMS Named Variables, each named **Reference_Num** and of type **ReferenceNum**. There shall be one MMS Named Variable for each entry in the ListOfLocalReference.

ListOfTransmissionSegment – sequence of MMS Named Variables. For each entry in the ListOfTransmissionSegment:

if the report is to contain periodic data:

for each Transmission Segment being reported:

An MMS Named Variable **TA_Transmssion_Segment**, type **TATransmissionSegment**

if floating point values are being reported for the segment:

An MMS Named Variable **Matrix_Id**, type **MatrixId** for each floating point quantity being reported. These define the meaning of each ‘column’ of data;

For each time period reported, an MMS Named Variable **Float_Array_XX**, type **FloatArrayXX**;

if integer values are being reported for the segment:

An MMS Named Variable **Matrix_Id**, type **MatrixId** for each integer quantity being reported. These define the meaning of each ‘column’ of data;

For each time period reported, an MMS Named Variable **Integer_Array_XX**, type **IntegerArrayXX**;

if the report is to contain profile data:

For each Transmission Segment being reported:

An MMS Named Variable **TA_Transmission_Segment_Profile**, type **TATransmissionSegmentProfile**

For each Profile Value being reported, an MMS Named Variable **TA_Profile_Value**, type **TAProfileValue**;

ListOfProfileValues – sequence of MMS Named Variables, each named **TA_Profile_Value**, and of type **TAProfileValue**, for each element in the ListOfProfileValues.

ListOfPeriodicValues – sequence of the following MMS Named Variables. Note that each of the FloatArray and IntegerArray variables below represents a distinct element in the ListOfPeriodicValues:

if floating point values are being reported for each period:
An MMS Named Variable **Matrix_Id**, type **MatrixId** for each floating point quantity being reported. These define the meaning of each 'column' of data;

For each time period reported, an MMS Named Variable **Float_Array_XX**, type **FloatArrayXX;**

if integer values are being reported for each period:
An MMS Named Variable **Matrix_Id**, type **MatrixId** for each integer quantity being reported. These define the meaning of each 'column' of data

For each time period being reported, an MMS Named Variable **Integer_Array_XX**, type **IntegerArrayXX**

See 6.7 for further description of the use of the Matrix types.

The order of reporting the MMS Named Variables representing the Transfer Account shall be as follows:

a) TAConditions_Detected variable (see 7.1.4.4.4 of IEC 60870-6-503);
b) Request_Id variable, if the report is in response to a Query Operation (see 7.1.5.1.2 of IEC 60870-6-503);
c) the Transfer Account header variable (see below for details);
d) variables representing the ListOfLocalReference, if any;
e) variables representing the ListOfTransmissionSegments, if any;
f) variables representing the ListOfProfileValues or ListOfPeriodicValues.

The MMS Named Variable (and its MMS Type) representing the header information is selected based on three attributes of the Transfer Account Object as defined in the following table:

TransmissionSegmentOption	DataType	MMS Variable
INCLUDED	PERIODIC	TA_Segments_Periodic
NOTINCLUDED	PERIODIC	TA_NoSegments_Periodic
INCLUDED	PROFILE	TA_Segments_Profile
NOTINCLUDED	PROFILE	TA_NoSegments_Profile

Other combinations of attributes are not supported in the Transfer Account Object Model.

The MMS Named Variables used have the following types:

TA_Segments_Periodic	type TASegmentsPeriodic
TA_NoSegments_Periodic	type TANoSegmentsPeriodic
TA_Segments_Profile	type TASegmentsProfile
TA_NoSegments_Profile	type TANoSegmentsProfile

The specific mapping of the Transfer Account Object attributes is as follows:

TransferAccountReference
Maps to the **TransferAccountRef** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**.

SendUtility
Maps to the **SendUtility** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**. Coding of specific values is determined by agreement between client and server.

ReceiveUtility
Maps to the **RecvUtility** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**. Coding of specific values is determined by agreement between client and server.

SellingAgent
Maps to the **SellingUtility** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**. Coding of specific values is determined by agreement between client and server.

BuyingAgent
Maps to the **BuyingUtility** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**. Coding of specific values is determined by agreement between client and server.

TimeStamp
Maps to the **TimeStamp** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**.

TransactionCode
Maps to the **TransactionCode** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**.

NumberOfLocalReference
Maps to the **NumberOfLocalReferences** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**.

ListOfLocalReference
Maps to a sequence of MMS Named Variables, each named **Reference_Num** and of type **ReferenceNum**. There shall be one MMS Named Variable reference in the report for each entry in the ListOfLocalReference. The number of Reference_Num references in the report shall agree with the value of the NumberOfLocalReference attribute.

Name
Maps to the **Name** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile**, or **TANoSegmentsProfile**. If the Name attribute is not present, the **Name** COMPONENT shall be the NULL string.

TransmissionSegmentOption
Not mapped, but used in selection of the data type of the MMS Variables (see above).

NumberOfTransSegments
Maps to the **NumberSegments** COMPONENT of an MMS Named Variable of type **TASegmentsPeriodic**, or **TASegmentsProfile**.

ListOfTransmissionSegment
Represented as a sequence of MMS Named Variables in the report. For each entry in the ListOfTransmissionSegment:

a) Report the segment definition parameters. If the report is to contain periodic data, then for each Transmission Segment being reported, report the segment definition parameters as the values of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment** (See 7.2.2.1 for details of this mapping). If the report is to contain profile data, then for each Transmission Segment being reported, report file segment definition parameters as the values of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile** (See 7.2.2.2 for details of this mapping).

b) Report the data for the segment:

If the report is to contain periodic data, then for each time period:

If any of the values being reported for the segment are of type floating point:

Report the identifiers of the floating point values in an MMS Named Variable with name **Matrix_Id** and type **MatrixId**

For each time period reported, report the floating point values for the time period relating to this segment in an MMS Named Variable with the name **Float_Array_XX** and type **FloatArrayXX**, where XX is large enough to contain all of the floating point values.

If any of the values are being reported for the segment are of type integer:

Report the identifiers of the integer values in an MMS Named Variable with name **Matrix_Id** and type **MatrixId**

For each time period reported, report the integer values for the time period relating to this segment in an MMS Named Variable with the name **Integer_Array_XX** and type **IntegerArrayXX**, where XX is large enough to contain all of the integer values.

If the report is to contain profile data, then for each time period being reported:

Report the profile data relating to this segment in an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

DataType
Not mapped, but used in selection of the data type of the MMS Variables (see above).

StartTime
Maps to the **StartTime** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, or **TANoSegmentsPeriodic**.

PeriodResolution
Maps to the **PeriodResolution** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, or **TANoSegmentsPeriodic**.

NumberOfPeriods
Maps to the **NumberPeriods** COMPONENT of an MMS Variable of type **TASegmentsPeriodic**, or **TANoSegmentsPeriodic**.

ListOfPeriodicValues
Represented as a sequence of MMS Named Variables in the report. For each entry in the ListOfPeriodicValues:

a) if floating point values are being reported:

Report the identifiers of the floating point values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each period report the values as the values of an MMS Named Variable with the name **Float_Array_XX**, where XX is large enough to contain all of the floating point values;

b) if integer values are being reported:

Report the identifiers of the integer values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each period report the values as the values of an MMS Named Variable with the name **Integer_Array_XX**, where XX is large enough to contain all of the integer values.

NumberOfProfiles
Maps to the **NumberOfProfiles** COMPONENT of an MMS Variable of type **TASegmentsProfile**, or **TANoSegmentsProfile**.

ListOfProfileValues
Maps to a sequence of MMS Named Variables, each named **TA_Profile_Value**, and of type **TAProfileValue**, for each element in the ListOfProfileValues. The number of **TA_Profile_Value** references in the report shall agree with the value of the **NumberOfProfiles** attribute.

7.2.2 TransmissionSegment Mapping

7.2.2.1 TATransmissionSegment

Each TransmissionSegment object maps to a sequence of MMS Named Variables. The first variable has name **TA_Transmission_Segment** and type **TATransmissionSegment**, and provides the parameters (reference identifiers, interchange points, etc.) for the segment. The remainder of the variables depend on the data types of the values being reported for each segment, and are described in the previous clause.

The segment definition attributes are mapped as follows:

TransmissionReference
Maps to the **TransmissionReference** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

UtilWheeling
Maps to the **UtilWheeling** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

UtilPaying
If present, maps to the **UtilityPaying** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

TransmissionSegType
Maps to the **TransmissionSegType** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**. The value is coded as follows: 0=DIRECT, 1=INONLY, 2=OUTONLY, 3=INOUT.

UtilIn
If present, maps to the **UtilIn** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

InterchangePtIn
If present, maps to the **InterchangePtIn** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

UtilOut
If present, maps to the **UtilOut** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

InterchangePtOut
If present, maps to the **InterchangePtOut** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

InterchangePt
If present, maps to the **InterchangePt** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment** and type **TATransmissionSegment**.

NumberFloatIds
The number of Float Ids in the **ListOfSegmentData** for this **TransmissionSegment.**

NumberIntegerIds
The number of Integer Ids in the **ListOfSegmentData** for this **TransmissionSegment.**

ListOfSegmentData
Mapped to MMS Named Variables of type **MatrixId**, **FloatArrayXX**, and **IntegerArrayXX**, as defined above.

7.2.2.2 TATransmissionSegmentProfile

Each Profile TransmissionSegment object maps to a sequence of MMS Named Variables. The first variable has name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**, and provides the parameters (reference identifiers, interchange points, etc.) for the segment. The remainder of the variables depend on the data types of the values being reported for each segment, and are described in the previous clause.

The segment definition attributes are mapped as follows:

TransmissionReference
Maps to the **TransmissionReference** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

UtilWheeling
Maps to the **UtilWheeling** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

UtilPaying
If present, maps to the **UtilityPaying** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

TransmissionSegType
Maps to the **TransmissionSegType** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**. The value is coded as follows: 0=DIRECT, 1=INONLY, 2=OUTONLY, 3=INOUT.

UtilIn
If present, maps to the **UtilIn** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

InterchangePtIn
If present, maps to the **InterchangePtIn** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

UtilOut
If present, maps to the **UtilOut** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

InterchangePtOut
If present, maps to the **InterchangePtOut** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

InterchangePt
If present, maps to the **InterchangePtIn** COMPONENT of an MMS Named Variable with name **TA_Transmission_Segment_Profile** and type **TATransmissionSegmentProfile**.

NumberProfileValues
The number of **ProfileValue** objects in the **ListofSegmentData** for this **TransmissionSegment**

ListOfSegmentData
Mapped to MMS Named Variables of type **ProfileValue** as defined above.

7.2.3 ProfileValue Mapping

RampStartTime
Maps to the **RampStartTime** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

RampDuration
Maps to the **RampDuration** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

ProfilePrice
Maps to the **ProfilePrice** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

TargetClass
Maps to the **ProfileTargetClass** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**. The value is encoded as: 0=OTHER, 1=ENERGY, 2=CAPACITY.

ProfileEnergy
If present, maps to the **ProfileTarget** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

ProfileCapacity
If present, maps to the **ProfileTarget** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

ProfileOther
If present, maps to the **ProfileTarget** COMPONENT of an MMS Named Variable with name **TA_Profile_Value** and type **TAProfileValue**.

7.2.4 AccountRequest Mapping

The AccountRequest object model maps to an MMS variable of type AccountRequest. The attributes map as:

AccountRequestName
Maps to the identifier of the MMS variable being written to generate the request.

TransferAccountReference
Maps to the **ReferenceTar** COMPONENT of an MMS variable of type **AccountRequest**.

StartTime
Maps to the **StartTime** COMPONENT of an MMS variable of type **AccountRequest**.

Duration
Maps to the **Duration** COMPONENT of an MMS variable of type **AccountRequest**.

RequestId
Maps to the **RequestId** COMPONENT of an MMS variable of type **AccountRequest**.

TaConditionsRequested
Maps to the **TaConditionsRequested** COMPONENT of an MMS variable of type **AccountRequest**.

7.3 Device Outage Mapping

This subclause defines the mapping of the Device Outage object model to MMS types. The Device Outage model is mapped to an MMS Named Variable for transmission using MMS Information Reports.

The Device Outage objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the Device Outage object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

The type and name of the MMS Named Variable depends on the Activity attribute of the Device Outage event. The following matrix defined the choice of name and type:

Activity	Type	Name
NEWPLAN	DONewRevSched	DO_New_Sched
REVISE	DONewRevSched	DO_Rev_Sched
CANCEL	DOCancel	DO_Cancel
ACTUAL	DOActual	DO_Actual

OutageReferenceId
Maps to the **OutageReferenceId** component of the selected data type.

OwningUtilityID
Maps to the **OwningUtilityId** component of the selected data type.

Timestamp
Maps to the **TimeStamp** component of the selected data type.

StationName
Maps to the **StationName** component of the selected data type.

DeviceType
Maps to the **DeviceType** component of the selected data type, with the following interpretation: 1=GENERATOR, 2=TRANSFORMER, 3=CAPACITOR, 4=TRANSMISSION_CIRCUIT, 5=BREAKER_SWITCH, 6=INDUCTOR, 0=OTHER.

DeviceName
Maps to the **DeviceName** component of the selected data type.

DeviceNumber
Maps to the **DeviceNumber** component of the selected data type.

DeviceRating
Maps to the **DeviceRating** component of the selected data type.

ActivityDateAndTime
Maps to the **ActivityDateAndTime** component of the selected data type.

Activity
Used in selecting the MMS Named Variable and type used in mapping the object.

PlanType
Maps to the **PlanType** component of an MMS Named Variable of type DONewRevSched, with the following interpretation: 0=SCHEDULED, 1=ESTIMATED.

PlannedOpenOrOutOfServiceDateAndTime
Maps to the **PlannedOpenOrOutOfServiceDateAndTime** component of an MMS Named Variable of type DONewRevSched.

PlannedCloseOrInServiceDateAndTime
Maps to the **PlannedCloseOrInServiceDateAndTime** component of an MMS Named Variable of type DONewRevSched.

OutagePeriod
Maps to the **OutagePeriod** of an MMS Named Variable of type DONewRevSched, with the following interpretation: 1=CONTINUOUS, 2=DAILY, 3=WEEKDAYS, 0=OTHER.

OutageType
Maps to the **OutageType** of an MMS Named Variable of type DONewRevSched, with the following interpretation: 0=FORCED, 1=MAINTENANCE, 2=PARTIAL, 3=ECONOMY, 4=UNPLANNED, 5=FORCED, 6=OTHER.

OutageAmountType
Maps to the **OutageAmountType** of an MMS Named Variable of type DONewRevSched, with the following interpretation: 0=PARTIAL, 1=FULL.

Amount
If present, maps to the **Amount** component of an MMS Named Variable of type DONewRevSched.

UpperOperatingLimit
If present, maps to the **UpperOperatingLimit** component of an MMS Named Variable of type DONewRevSched.

LowerOperatingLimit
If present, maps to the **LowerOperatingLimit** component of an MMS Named Variable of type DONewRevSched.

Class
If present, maps to the **Class** component of an MMS Named Variable of type DONewRevSched with the following interpretation: 0=OUTSERVICE, 1=INSERVICE.

Action
If present, maps to the **Action** component of an MMS Named Variable of type DOActual with the following interpretation: 0=TRIPPED, 1=OFFLINE, 2=ONLINE, 3=OPEN, 4=CLOSE.

Affected Amount
If present, maps to the **AffectedAmount** component of an MMS Named Variable of type DOActual.

Comments
Maps to the **Comments** component of the selected data type.

OutageEffect
Maps to the **OutageEffect** component of the selected data type.

7.4 Information Buffer Mapping

The Information Buffer object maps onto two MMS Named Variables. The first variable is of type **InfoMessHeader**, and contains the global information about the message. The second variable is of type **InfoBufXX**, where XX is large enough to hold the entire message. The mapping of the attributes is as follows:

InfoReference
Maps onto the **InfoReference** COMPONENT of an MMS Named Variable of type **InfoMessHeader**.

LocalReference
Maps onto the **LocalReference** COMPONENT of an MMS Named Variable of type **InfoMessHeader**.

MessageId
Maps onto the **MessageId** COMPONENT of an MMS Named Variable of type **InfoMessHeader**.

Size
Maps onto the **Size** COMPONENT of an MMS Named Variable of type **InfoMessHeader**.

InfoStream
Maps onto an MMS Named Variable of type **InfoBufXX**, where XX is large enough to hold all of the data. Note that XX can be larger, since the **Size** attribute determines how much of the buffer is actually valid.

7.5 Power Plant Mapping

7.5.1 Availability Report Mapping

The MMS Named Variable (and its MMS Type) representing the Availability Report is selected based on the Availability Status attribute of the Availability Report Object. If the Availability Status attribute is AVAILABLE, the report maps onto an MMS Named Variable of name **Available** and type **Available**. If the Availability Status attribute is UNAVAILABLE, the report maps onto an MMS Named Variable of name **UnAvailable** and type **UnAvailable**.

The Availability Report objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated.

Each of the MMS Named Variables used to map the Availability Report object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

The remainder of the object attributes map as follows:

AvailabilityReferenceID
Maps to the **AvailabilityReferenceId** COMPONENT of an MMS Named Variable of the selected name and type.

Timestamp
Maps to the **Timestamp** COMPONENT of an MMS Named Variable of the selected name and type.

PlantReferenceID
Maps to the **PlantReferenceID** COMPONENT of an MMS Named Variable of the selected name and type.

UnitID
Maps to the **UnitID** COMPONENT of an MMS Named Variable of the selected name and type.

ReportStatus
Maps to the **ReportStatus** COMPONENT of an MMS Named Variable of the selected name and type with the following interpretation: 0=PROPOSED, 1=CONFIRMED, 2=CANCELLED.

StartDateAndTime
Maps to the **StartDateAndTime** COMPONENT of an MMS Named Variable of the selected name and type.

StopDateAndTime
Maps to the **StopDateAndTime** COMPONENT of an MMS Named Variable of the selected name and type.

Duration
Maps to the **Duration** COMPONENT of an MMS Named Variable of the selected name and type.

Availability Status
The Availability Status attribute is used in the selection of the name and type of the MMS Named Variable representing the report.

EconomicImpact
Maps to bit 0 of the **Impact** COMPONENT of the MMS Named Variable of type **Available** with the following interpretation: 1=YES, 0=NO.

PriceImpact
If present, maps to the **PriceImpact** COMPONENT of the MMS Named Variable of type **Available**. If bit 0 of the **Impact** COMPONENT is NO, the **PriceImpact** COMPONENT shall be ignored.

RampRateImpact
Maps to bit 1 of the **Impact** COMPONENT of the MMS Named Variable of type **Available** with the following interpretation: 1=YES, 0=NO.

MaxRampRateUp
If present, maps to the **MaxRampRateUp** field within the **RampImpact** COMPONENT of the MMS Named Variable of type **Available**. If bit 1 of the **Impact** COMPONENT is NO, the **MaxRampRateUp** COMPONENT shall be ignored.

MaxRampRateDown
If present, maps to the **MaxRampRateDown** field within the **RampImpact** COMPONENT of the MMS Named Variable of type **Available**. If bit 1 of the **Impact** COMPONENT is NO, the **MaxRampRateDown** COMPONENT shall be ignored.

CapacityImpact
Maps to bit 2 of the **Impact** COMPONENT of the MMS Named Variable of type **Available** with the following interpretation: 1=YES, 0=NO.

Each of the following attributes (if present) map into fields within the **CapacityImpact** COMPONENT (type **UnitCapacity**) of the MMS Named Variable of type **Available**. If bit 2 of the **Impact** COMPONENT is NO, all of the fields within the **CapacityImpact** COMPONENT shall be ignored.

UnitCapacity
If present, maps onto the **CapacityClass** COMPONENT of the type **UnitCapacity** with the following interpretation: bit 0 = 1 implies GROSS, bit 1 = 1 implies NET, both bit 0 and bit 1 = 1 implies BOTH.

GrossMaxCapacity
If present, maps onto the **GrossMaxCapacity** COMPONENT of the type **UnitCapacity**. If bit 0 of the **CapacityClass** COMPONENT is not = 1, the **GrossMaxCapacity** shall be ignored.

GrossMinCapacity
If present, maps onto the **GrossMinCapacity** COMPONENT of the type **UnitCapacity**. If bit 0 of the **CapacityClass** COMPONENT is not = 1, the **GrossMinCapacity** shall be ignored.

NetMaxCapacity
If present, maps onto the **NetMaxCapacity** COMPONENT of the type **UnitCapacity**. If bit 1 of the **CapacityClass** COMPONENT is not = 1, the **NetMaxCapacity** shall be ignored.

NetMinCapacity

If present, maps onto the **NetMinCapacity** COMPONENT of the type **UnitCapacity**. If bit 1 of the **CapacityClass** COMPONENT is not = 1, the **NetMinCapacity** shall be ignored.

Each of the following attributes (when present) map into fields within the **Availability** COMPONENT (type **AvailabilityClass**) of the MMS Named Variable of type **Available**:

TypeOfAvailability
Maps to bit 0 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=STANDBY, 1=ONLINE.

TimeToOnline
Maps to the **TimeToOnline** COMPONENT of the type **AvailabilityClass**. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 1, this attribute shall be ignored.

LFC
Maps to bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES.

Dispatchable
Maps to bit 2 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

Regulating
Maps to bit 3 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

Manually_Loaded
Maps to bit 4 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

ReasonForNoLFC
Maps to the **ReasonForNoLFC** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=STARTUP, 1=UNSTABLE. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 1, this attribute shall be ignored.

ReasonForUnavailable
Maps to the **ReasonForUnavailable** COMPONENT of the MMS Named Variable of type **UnAvailable** with the following interpretation: 0=FORCED, 1=SCHEDULED, 2=TESTING.

ProvidingReserve
Maps to the **ProvidingReserve** COMPONENT of the selected MMS Named Variable (type **Available** or **UnAvailable**) with the following interpretation: 0=NO, 1=YES.

Comment
Maps to the **Comment** COMPONENT of the selected MMS Named Variable (type **Available** or **UnAvailable**).

7.5.2 Real Time Status Mapping

The MMS Named Variable (and its MMS Type) representing the Real Time Status Report is selected based on the Availability Status attribute of the Real Time Status Object. If the Availability Status attribute is AVAILABLE, the report maps onto an MMS Named Variable of name **Status_Available** and type **StatusAvailable**. If the Availability Status attribute is UNAVAILABLE, the report maps onto an MMS Named Variable of name **Status_UnAvailable** and type **StatusUnAvailable**.

The Real Time Status objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the Real Time Status object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

The remainder of the object attributes map as follows:

AvailabilityReferenceID
Maps to the **AvailabilityReferenceID** COMPONENT of an MMS Named Variable of the selected name and type.

Timestamp
Maps to the **Timestamp** COMPONENT of an MMS Named Variable of the selected name and type.

PlantReferenceID
Maps to the **PlantReferenceID** COMPONENT of an MMS Named Variable of the selected name and type.

UnitID
Maps to the **UnitID** COMPONENT of an MMS Named Variable of the selected name and type.

Availability Status
The Availability Status attribute is used in the selection of the name and type of the MMS Named Variable representing the report.

MaxRampRateUp
If present, maps to the **MaxRampRateUp** field within the **RampStatus** COMPONENT of the MMS Named Variable of type **StatusAvailable**.

MaxRampRateDown
If present, maps to the **MaxRampRateDown** field within the **RampStatus** COMPONENT of the MMS Named Variable of type **StatusAvailable**.

Each of the following attributes (if present) map into fields within the **CapacityStatus** COMPONENT (type **UnitCapacity**) of the MMS Named Variable of type **StatusAvailable**.

UnitCapacity
If present, maps onto the **CapacityClass** COMPONENT of the type **UnitCapacity** with the following interpretation: bit 0 = 1 implies GROSS, bit 1 = 1 implies NET, both bit 0 and bit 1 = 1 implies BOTH.

GrossMaxCapacity
If present, maps onto the **GrossMaxCapacity** COMPONENT of the type **UnitCapacity**. If bit 0 of the **CapacityClass** COMPONENT is not = 1, the **GrossMaxCapacity** shall be ignored.

GrossMinCapacity
If present, maps onto the **GrossMinCapacity** COMPONENT of the type **UnitCapacity**. If bit 0 of the **CapacityClass** COMPONENT is not = 1, the **GrossMinCapacity** shall be ignored.

NetMaxCapacity
If present, maps onto the **NetMaxCapacity** COMPONENT of the type **UnitCapacity**. If bit 1 of the **CapacityClass** COMPONENT is not = 1, the **NetMaxCapacity** shall be ignored.

NetMinCapacity
If present, maps onto the **NetMinCapacity** COMPONENT of the type **UnitCapacity**. If bit 1 of the **CapacityClass** COMPONENT is not = 1, the **NetMinCapacity** shall be ignored.

Each of the following attributes (when present) map into fields within the **AvailabilityStatus** COMPONENT (type **AvailabilityClass**) of the MMS Named Variable of type **StatusAvailable**:

TypeOfAvailability
Maps to bit 0 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=STANDBY, 1=ONLINE.

TimeToOnline
Maps to the **TimeToOnline** COMPONENT of the type **AvailabilityClass**. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 1, this attribute shall be ignored.

LFC
Maps to bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES.

Dispatchable
Maps to bit 2 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

Regulating
Maps to bit 3 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

Manually_Loaded
Maps to bit 4 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=NO, 1=YES. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 0, this attribute shall be ignored.

ReasonForNoLFC
Maps to the **ReasonForNoLFC** COMPONENT of the type **AvailabilityClass** with the following interpretation: 0=STARTUP, 1=UNSTABLE. If bit 1 of the **AvailFlags** COMPONENT of the type **AvailabilityClass** is 1, this attribute shall be ignored.

ReasonForUnavailable
Maps to the **ReasonForUnavailable** COMPONENT of the MMS Named Variable of type **StatusUnAvailable** with the following interpretation: 0=FORCED, 1=SCHEDULED, 2=TESTING, 3=EQUIPMENT.

ProvidingReserve
Maps to the **ProvidingReserve** COMPONENT of the selected MMS Named Variable (type **StatusAvailable** or **StatusUnAvailable**) with the following interpretation: 0=NO, 1=YES.

7.5.3 Forecast Mapping

The Forecast Report Object is mapped onto an MMS Named Variable with name **Forecast** and type **Forecast**.

The Forecast objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the Forecast object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

The object attributes map as follows:

ForecastScheduleReferenceID
Maps to the **ForecastScheduleReferenceID** COMPONENT of an MMS Named Variable named **Forecast**.

PlantReferenceID
Maps to the **PlantReferenceID** COMPONENT of an MMS Named Variable named **Forecast**.

UnitID
Maps to the **UnitID** COMPONENT of an MMS Named Variable named **Forecast**.

ForecastType
Maps to the **ForecastType** COMPONENT of an MMS Named Variable named **Forecast** with the following interpretation: bit 0 set to 1 for GENERATION, bit 1 set to 1 for RESERVE, both bits 0 and 1 set to 1 for BOTH.

StartTime
Maps to the **StartTime** COMPONENT of an MMS Named Variable named **Forecast**.

PeriodResolution
Maps to the **PeriodResolution** COMPONENT of an MMS Named Variable named **Forecast**.

NumberOfPeriods
Maps to the **NumberOfPeriods** COMPONENT of an MMS Named Variable named **Forecast**.

ListOfForecasts
Represented as a sequence of MMS Named Variables in the report. For each entry in the ListOfForecasts:

a) report the identifier of the floating point value **MW** as the value of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

b) for each time period reported, report the MW value as the value of an MMS Named Variable **Float_Array_1** and type **FloatArray1**;

c) report the identifier of the integer value **LFC_Code** as the value of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

d) for each time period being reported, report the LFC_Code value as the value of an MMS Named Variable with name **Integer_Array_1** and type **IntegerArray1**.

7.5.4 Curve Mapping

Curve Objects are mapped to a sequence of MMS Named Variables. The first variable (name **Curve**, type **Curve**) defines the global information about the curve: name, reference identifiers, curve type, and number of curve segments. Each segment of the curve is represented as:

a) an MMS Named Variable (name **Curve_Segment**, type **CurveSegmentDescription**);

b) an MMS Named Variable (name **Float_Array_XX**, type **FloatArrayXX**, where XX is the number of coefficients used to describe the curve segment.

The Curve objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the Curve object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

The curve attributes map as follows:

CurveName
Maps to the **CurveName** COMPONENT of an MMS Named Variable of type **Curve**.

PlantReferenceID
Maps to the **PlantReferenceID** COMPONENT of an MMS Named Variable of type **Curve**.

UnitID
Maps to the **UnitID** COMPONENT of an MMS Named Variable of type **Curve**.

CurveType
Maps to the **CurveType** COMPONENT of an MMS Named Variable of type **Curve**.

NumberOfSegments
Maps to the **NumberOfSegments** COMPONENT of an MMS Named Variable of type **Curve**.

SequenceOfCurveSegmentDescription
Maps to an MMS Named Variable of name **Curve_Segment** and type **CurveSegmentDescription**. The mapping of the curve segment attributes is as follows:

Order
Maps to the **Order** COMPONENT of an MMS Named Variable of type **CurveSegmentDescription**.

LowRange
Maps to the **LowRange** COMPONENT of an MMS Named Variable of type **CurveSegmentDescription**.

HighRange
Maps to the **HighRange** COMPONENT of an MMS Named Variable of type **CurveSegmentDescription**.

SequenceOfCoefficients
Maps to the elements of an MMS Named Variable of type **FloatArrayXX**, where XX is the value of the **Order** attribute.

7.6 General Data Report Mapping

General Data Reports are only visible through the Transfer Account Reporting mechanism as defined in IEC 60870-6-503. The General Data Reports are mapped to a sequence of one or more MMS Named Variables but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the General Data Report object model are only used in the MMS Information Reports and return the MMS Access Result OBJECT_ACCESS_DENIED when read or written.

Each General Data Report contains header information and optionally one or more associated lists (ListOfLocalReference, ListOfFloatingPoint1Values, ListOfFloatingPoint2Values, ListOf Integer1Values, ListOfInteger2Values, ListOfText1Values, ListOfText2Values). The list of data is represented as follows:

ListOfLocalReference – sequence of MMS Named Variables, each named **Reference_Num** and of type **ReferenceNum**. There shall be one MMS Named Variable for each entry in the ListOfLocalReference.

ListOfFloatingPoint1Values/ListOfFloatingPoint2Values – sequence of the following MMS Named Variables. (Note that each of the FloatArray variables below represents a distinct element in the ListOfFloatingPoint1Values or the ListOfFloatingPoint2Values.)

An MMS Named Variable **Matrix_Id**, type **MatrixId** for each floating point quantity being reported. These defined the meaning of each column of floating point data.

An MMS Named Variable **Float _Array_XX**, type **FloatArrayXX**.

ListOfInteger1Values/ListOfInteger2Values – sequence of the following MMS Named Variables. (Note that each of the IntegerArray variables below represents a distinct element in the ListOfInteger1Values or ListOfInteger2Values.)

An MMS Named Variable **Matrix_Id**, type **MatrixId** for each integer quantity being reported. These defined the meaning of each column of integer data.

An MMS Named Variable **Integer_Array_XX**, type **IntegerArrayXX**.

ListOfText1Values/ListOfText2Values – sequence of the following MMS Named Variables. (Note that each of the Text32Array variables below represents a distinct element in the ListOfText1Values or the ListOfText2Values.)

An MMS Named Variable **Matrix_Id**, type **MatrixId** for each text quantity being reported. These defined the meaning of each column of text data.

An MMS Named Variable **Text32_Array_XX**, type **Text32ArrayXX**.

The order of reporting the MMS Named Variables representing the GeneralDataReport shall be as follows:

a) TAConditions_Detected variable (see 60870-6-503, clause 7.1.4.4)
b) Request_Id variable, if the report is in response to a Query Operation (see 60870-6-503, clause 7.1.5.1.2).
c) The General_Data_Report variable of type GeneralDataReport representing the GeneralDataReport header
d) Variables representing the ListOfLocalReference, if any
e) Variables representing the ListOfFloatingPoint1Values, if any
f) Variables representing the ListOfFloatingPoint2Values, if any
g) Variables representing the ListOfInteger1Values, if any
h) Variables representing the ListOfInteger2Values, if any
i) Variables representing the ListOfText1Values, if any
j) Variables representing the ListOfText2Values, if any

The specific mapping of the General Data Report object attributes are as follows:

GeneralDataReportReferenceNumber
Maps to the **GeneralDataReportReferenceNumber** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ReportName
Maps to the **ReportName** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ReportDateAndTime
Maps to the ReportDateAndTime COMPONENT of an MMS Variable of type GeneralDataReport.

TransactionCode
Maps to the **TransactionCode** COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfLocal Reference
Maps to the **NumberOfLocalReference** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfLocalReference
Maps to a sequence of MMS Named Variables, each named **Reference_Num** and of type **ReferenceNum**. There shall be one MMS Named Variable reference in the report for each entry in the **ListOfLocalReference**. The number of **Reference_Num** references in the report shall agree with the value of the **NumberOfLocalReference** attribute.

NumberOfFloatingPoint1
Maps to the **NumberOfFloatingPoint1** COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfFloatingPoint1Rows
Maps to the **NumberOfFloatingPoint1Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfFloatingPoint1Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfFloatingPoint1Values**:

Report the identifiers of the floating point values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Float_Array_XX**, where XX is large enough to contain all of the floating point values.

NumberOfFloatingPoint2
Maps to the **NumberOfFloatingPoint2** COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfFloatingPoint2Rows
Maps to the **NumberOfFloatingPoint2Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfFloatingPoint2Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfFloatingPoint2Values**:

Report the identifiers of the floating point values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Float_Array_XX**, where XX is large enough to contain all of the floating point values.

NumberOfInteger1
Maps to the **NumberOfInteger1**COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfInteger1Rows
Maps to the **NumberOfInteger1Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfInteger1Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfInteger1Values**:

Report the identifiers of the integer values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Integer_Array_XX**, where XX is large enough to contain all of the floating point values.

NumberOfInteger2
Maps to the **NumberOfInteger2**COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfInteger2Rows
Maps to the **NumberOfInteger2Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfInteger2Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfInteger1Values**:

Report the identifiers of the integer values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Integer_Array_XX**, where XX is large enough to contain all of the floating point values.

NumberOfText1
Maps to the **NumberOfText1**COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfText1Rows
Maps to the **NumberOfText1Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfText1Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfText1Values**:

Report the identifiers of the text values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Text32_Array_XX**, where XX is large enough to contain all of the floating point values.

NumberOfText2
Maps to the **NumberOfText2**COMPONENT of an MMS Variable of type **GeneralDataReport**.

NumberOfText2Rows
Maps to the **NumberOfText2Rows** COMPONENT of an MMS Variable of type **GeneralDataReport**.

ListOfText2Values
Represented as a sequence of MMS Named Variables in the report. For each entry in the **ListOfText2Values**:

Report the identifiers of the text values as the values of an MMS Named Variable with name **Matrix_Id** and type **MatrixId**;

For each row report the values as the values of an MMS Named Variable with the name **Text32_Array_XX**, where XX is large enough to contain all of the floating point values.

7.7 General Data Response Mapping

The General Data Response Object is mapped onto an MMS Named Variable with name **General_Data_Response** and type **GeneralDataResponse**.

The General Data Response objects are mapped to a sequence of one or more MMS Named Variables, but only instantaneously while the report is being generated. Each of the MMS Named Variables used to map the General Data Response object model are only used in MMS Information Reports, and return the MMS Access Result OBJECT-ACCESS-DENIED when read or written.

Each General Data Response object contains header information and optionally ListOfLocalReference, where:

ListOfLocalReference – sequence of MMS Named Variables, each named **Reference_Num** and of type **ReferenceNum**. There shall be one MMS Named Variable for each entry in the **ListOfLocalReference**.

The specific mapping of the General Data Response object attributes are as follows:

ReportReferenceNumber
Maps to the **ReportReferenceNumber** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

ReportName
Maps to the **ReportName** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

ReportTimeStamp
Maps to the **ReportTimeStamp** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

NumberOfLocal Reference
Maps to the **NumberOfLocalReference** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

ResponseData
Maps to the **ResponseData** COMPONENT of an MMS Variable of type **GeneralDataResponse**

ResponseCode
Maps to the **ResponseCode** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

ResponseText
Maps to the **ResponseText** COMPONENT of an MMS Variable of type **GeneralDataResponse**.

8 Use of Supervisory Control Objects

The Supervisory Control object models (IndicationPoint and ControlPoint) are generic in nature in that more than one type of device can be represented with these object models. This clause provides the allowable uses of these object models to represent real devices. However, it is recognized that this list may not be exhaustive. If a new device is defined in the future that requires different semantics (i.e. interpretations) that cannot be mapped into the existing list, then implementers can add new semantics as long as they do not conflict with the existing semantics assigned to values in this section.