

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

**Matériels et systèmes de téléconduite –
Partie 6-802: Protocoles de téléconduite compatibles avec les normes ISO
et les recommandations de l'UIT-T – Modèles d'objets TASE.2**



THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2014 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.



IEC 60870-6-802

Edition 3.0 2014-07

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

**Matériels et systèmes de téléconduite –
Partie 6-802: Protocoles de téléconduite compatibles avec les normes ISO
et les recommandations de l'UIT-T – Modèles d'objets TASE.2**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE **XD**
CODE PRIX

ICS 33.200

ISBN 978-2-8322-1652-1

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms and definitions	7
4 Abbreviations	7
5 Object models	7
5.1 General.....	7
5.2 Supervisory Control and Data Acquisition	8
5.2.1 General	8
5.2.2 IndicationPoint object	8
5.2.3 ControlPoint Object	11
5.2.4 Protection Equipment Event Object Model	13
5.3 Device Outage Object.....	16
5.4 InformationBuffer Object	19
6 MMS Types for Object Exchange.....	19
6.1 General.....	19
6.2 Supervisory Control and Data Acquisition Types	20
6.2.1 IndicationPoint Type Descriptions	20
6.2.2 ControlPoint Type Descriptions.....	23
6.2.3 Protection Equipment Type Descriptions.....	23
6.3 Device Outage Type Descriptions	24
6.4 InformationBuffer Type Descriptions	26
7 Mapping of Object Models to MMS Types	26
7.1 Supervisory Control and Data Mapping	26
7.1.1 Indication Object Mapping	26
7.1.2 ControlPoint Object Mapping	29
7.1.3 Protection Event Mapping.....	30
7.2 Device Outage Mapping	33
7.3 Information Buffer Mapping.....	35
8 Use of Supervisory Control Objects	36
8.1 General.....	36
8.2 Use of IndicationPoint Model.....	36
8.3 Use of ControlPoint Model	37
9 Conformance.....	37
Annex A (informative) TASE.2 (2002) Additional Object Models	39
A.1 General.....	39
A.2 Transfer Accounts.....	39
A.3 Power Plant Objects	46
A.3.1 General	46
A.3.2 Availability Report Object	46
A.3.3 Real Time Status Object.....	50
A.3.4 Forecast Schedule Object.....	53
A.4 General Data Report Object.....	55
A.4.1 General	55

A.4.2	General Data Request Object	56
A.4.3	General Data Response Object	59
Annex B (informative)	TASE.2 (2002) Additional MMS Object Types	61
B.1	General	61
B.2	Transfer Account Types	61
B.3	Power Plant Type Descriptions	63
B.4	Power System Dynamics	66
B.4.1	General	66
B.4.2	Matrix Data Types	67
B.5	GeneralDataReport Type Descriptions	68
B.6	GeneralDataResponse Type Descriptions	68
Annex C (informative)	TASE.2 (2002) Mapping of Objects to MMS Types	69
C.1	General	69
C.2	Transfer Accounts Mapping	69
C.2.1	TransferAccount Mapping	69
C.2.2	TransmissionSegment Mapping	73
C.2.3	ProfileValue Mapping	76
C.2.4	AccountRequest Mapping	76
C.3	Power Plant Mapping	77
C.3.1	Availability Report Mapping	77
C.3.2	Real Time Status Mapping	80
C.3.3	Forecast Mapping	82
C.3.4	Curve Mapping	83
C.4	General Data Report Mapping	85
C.4.1	General Data Request Mapping	85
C.4.2	General Data Response Mapping	88
Annex D (informative)	Transfer account examples	90

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 itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 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 systems management and associated information exchange.

This third edition cancels and replaces the second edition published in 2002 and its amendment 1 (2005). This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) Accounts, Programs, Event Enrollment and Event Condition objects have been changed from informative to normative. As a result, the conformance tables have been updated.
- b) The services associated with Accounts, Programs, Event Enrollment and Event Conditions are now out of scope.
- c) The TASE.2 conformance blocks 6, 7, 8 and 9 have been made out of scope.

- d) The MMS Mappings for Accounts, Programs, Event Enrollment and Event Condition objects have been changed from normative to informative.

The text of this standard is based on the following documents:

FDIS	Report on voting
57/1455/FDIS	57/1479/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 60870 series, published under the general title *Telecontrol equipment and systems*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability 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 is 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. 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 would be advisable.

The notation of the object modelling used for the objects specified in Clause B.2 is defined in IEC 60870-6-503. This part of IEC 60870-6 is based on the TASE.2 services and protocol. To understand the modelling and semantics of this part of IEC 60870-6, some basic knowledge of IEC 60870-6-503 would be advisable.

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.

Clause 9 is the standards conformance table.

An informative Annex A 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 documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60870-5-101:2003, *Telecontrol equipment and systems – Part 5-101: Transmission protocols – Companion standard for basic telecontrol tasks*

IEC 60870-6-503:2014, *Telecontrol equipment and systems – Part 6-503: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Services and protocol*

ISO 9506-1:2003, *Industrial automation systems – Manufacturing Message Specification – Part 1: Service definition*

ISO 9506-2:2003, *Industrial automation systems – Manufacturing Message Specification – Part 2: Protocol specification*

3 Terms and definitions

For the purposes of this part of IEC 60870, the terms and 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

5.1 General

Object models are required for various functions within a system. Clause 5 delineates abstract object models based on functionality. Object models within one functional area may be used in another functional area.

5.2 Supervisory Control and Data Acquisition

5.2.1 General

The object models in this clause are derived from the historical perspective of Supervisory Control and Data Acquisition (SCADA) systems. This subclause 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.

5.2.2 IndicationPoint object

An IndicationPoint object represents an actual input point.

Object: **IndicationPoint** (Read Only)

Key Attribute: PointName

Attribute: PointType (REAL, STATE, DISCRETE, STATESUPPLEMENTAL)

Constraint PointType=REAL

Attribute: PointRealValue

Constraint PointType=STATE

Attribute: PointStateValue

Constraint PointType=DISCRETE

Attribute: PointDiscreteValue

Constraint PointType= STATESUPPLEMENTAL

Attribute: PointStateSupplementalValue

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, STATESUPPLEMENTAL.

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.

PointStateSupplementalValue

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

A PointStateSupplementalValue shall have the ability to indicate the current value (State), tagging information (Tag), and the expected value/state (ExpectedState). If the ExpectedState value does not match the State value, this indicates that the provider of the ExpectedState value is indicating a potential issue.

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 `NOTTIMESTAMP` 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`, `PointDiscreteValue`, or `PointStateSupplementalValue`) 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, PointDiscreteValue, or PointStateSupplementalValue) 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 **C**ounter) 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, PointDiscreteValue, or PointStateSupplementalValue) of the IndicationPoint has changed. It is incremented each time the owner sets a new value for the IndicationPoint.

5.2.3 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.

5.2.4 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.

5.3 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: Messageld

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.).

Messageld

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.

6 MMS Types for Object Exchange

6.1 General

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.

6.2 Supervisory Control and Data Acquisition Types

6.2.1 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_StateSupplemental bit-string:

```
{
    State_hi[0],
    State_lo[1],
    Tag_hi[2],
    Tag_lo[3],
    ExpectedState_hi[4],
    ExpectedState_lo[5],
    Reserved[6],
    Reserved[7]
}
```

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 STRUCTURE
{
    COMPONENT Value          Data_State,
    COMPONENT Flags          Data_Flags
}

Data_DiscreteQ STRUCTURE
{
    COMPONENT Value          Data_Discrete,
    COMPONENT Flags          Data_Flags
}

Data_StateSupplementalQ STRUCTURE
{
    COMPONENT Value          Data_StateSupplemental,
    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_StateSupplementalQTimeTag STRUCTURE
{
    COMPONENT Value          Data_StateSupplemental,
    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_StateSupplementalExtended STRUCTURE
{
    COMPONENT Value          Data_StateSupplemental,
    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
}
Data_State_SupplementalQTimeTagExtended STRUCTURE
{
    COMPONENT Value          Data_StateSupplemental,
    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 }
}

```


6.2.2 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 }
}
```

6.2.3 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         TimeInterval16,
    COMPONENT EventTime              TimeStampExtended
}

```

PackedProtectionEvent STRUCTURE

```

{
    COMPONENT PackedEvent           EventFlags,
    COMPONENT PackedEventFlags      PackedFlags,
    COMPONENT OperatingTime         TimeInterval16,
    COMPONENT EventTime              TimeStampExtended
}

```

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}
OutageAmountTypeld	integer {width 16}
OutagePeriodId	integer {width 16}
OutageTypeld	integer {width 16}
PlanTypeld	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             Deviceld,
    COMPONENT DeviceNumber         Number,
    COMPONENT DeviceRating         Data_Real,
    COMPONENT ActivityDateAndTime   ScheduleTime,
    COMPONENT PlanType             PlanTypeld,
    COMPONENT PlannedOpenOrOutOfServiceDateAndTime ScheduleTime,
    COMPONENT PlannedCloseOrInServiceDateAndTime ScheduleTime,
    COMPONENT OutagePeriod         OutagePeriodId,
    COMPONENT OutageType           OutageTypeld,
    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             Deviceld,
    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             Deviceld,

```

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.

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
STATE SUPPLEMENTAL	NOQUALITY	NOTIMESTAMP	Data_StateSupplemental
REAL	QUALITY	NOTIMESTAMP	Data_RealQ
STATE	QUALITY	NOTIMESTAMP	Data_StateQ
DISCRETE	QUALITY	NOTIMESTAMP	Data_DiscreteQ
STATE SUPPLEMENTAL	QUALITY	NOTIMESTAMP	Data_StateSupplementalQ
REAL	QUALITY	TIMESTAMP	Data_RealQTimeTag
STATE	QUALITY	TIMESTAMP	Data_StateQTimeTag
DISCRETE	QUALITY	TIMESTAMP	Data_DiscreteQTimeTag
STATE SUPPLEMENTAL	QUALITY	TIMESTAMP	Data_StateSupplementalQTimeTag
REAL	QUALITY	TIMESTAMPEXTENDED	Data_RealQTimeTagExtended
STATE	QUALITY	TIMESTAMPEXTENDED	Data_StateQTimeTagExtended
DISCRETE	QUALITY	TIMESTAMPEXTENDED	Data_DiscreteQTimeTagExtended
STATE SUPPLEMENTAL	QUALITY	TIMESTAMPEXTENDED	Data_StateSupplementalQTimeTag Extended

If COVClass is COV, the following criteria are used:

PointType	Map to type:
REAL	Data_RealExtended
STATE	Data_StateExtended
DISCRETE	Data_DiscreteExtended
STATESUPPLEMENTAL	Data_StateSupplementalExtended

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, 3=StateSupplemental.

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.

PointStateSupplementalValue

If present, maps to either the value of the MMS variable of type **Data_StateSupplemental** (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 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.

7.3 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.

8 Use of Supervisory Control Objects

8.1 General

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 Clause 8.

8.2 Use of IndicationPoint Model

The IndicationPoint model is used to represent arbitrary data input from devices such as status points (PointType=STATE, PointType=STATESUPPLEMENTAL, or PointType=DISCRETE), analog points (PointType=REAL) and counter values (PointType=DISCRETE), and Transformer step positions (PointType=DISCRETE).

PointType STATE and STATESUPPLEMENTAL are recommended for status points (single or double) with up to three states whereas PointType DISCRETE is recommended for status points with more than three states. The following PointValue values of type STATE are used to represent specific device positions:

00	01	10	11	Device
Between	Tripped	Closed	Invalid	Disconnecter
Between	Off	On	Invalid	Disconnecter
Invalid	Off	On	Invalid	Breaker
Invalid	Auto	Manual	Invalid	
Invalid	Normal	Alarm	Invalid	
Invalid	Local	Remote	Invalid	
Invalid	Raise	Lower	Invalid	
Invalid	Not Ready	Ready	Invalid	
Invalid	Offline	Available	Invalid	

If DISCRETE is used for single and double point information, the following PointValue values of type DISCRETE (integer) are used to represent specific device positions:

0	1	2	3	Device
Between	Tripped	Closed	Invalid	Disconnecter
Between	Off	On	Invalid	Disconnecter
Invalid	Off	On	Invalid	Breaker
Invalid	Auto	Manual	Invalid	
Invalid	Normal	Alarm	Invalid	
Invalid	Local	Remote	Invalid	
Invalid	Raise	Lower	Invalid	
Invalid	Not Ready	Ready	Invalid	
Invalid	Offline	Available	Invalid	

Analog inputs may be represented as either PointType=REAL (if scaling and normalization procedures are done at the TASE.2 server end) or as PointType=DISCRETE (if raw values are being input).

8.3 Use of ControlPoint Model

The ControlPoint model is used to represent arbitrary data output as switching commands to devices such as switching devices and transformers (ControlPointType=COMMAND), analog and digital setpoints to devices or units such as power units (ControlPointType=SETPOINT). The following values of type COMMAND (integer) are used to represent specific device switching commands:

0	1	Device
Trip	Close	Switch
Open	Close	Switch
Off	On	Switch
Lower	Raise	Transformer

Analog outputs may be modelled as either ControlPointType=REAL (if scaling and normalization procedures are done at the TASE.2 server end) or as ControlPointType=DISCRETE (if raw values are specified).

9 Conformance

The object models in this part of IEC 60870 have been grouped according to the service conformance blocks as defined in Clause 9 of IEC 60870-6-503:2014. The following tables define in detail the conformance requirements of TASE.2 implementations. Throughout these tables, the entry O implies optional, and Mⁿ implies that the construction is mandatory for conformance block n, as defined below. "i" indicates the objects to be out-of-scope of normative part of this document.

Supervisory Control and Data Acquisition	Client	Server
IndicationPoint Object	M ¹	M ¹
IndicationPoint Object (STATESUPPLEMENTAL)	O ¹	O ¹
ControlPoint Object	M ⁵	M ⁵
ProtectionEquipmentEvent Object	O	O

Transfer Accounts	Client	Server
TransferAccount Object	i ⁸	i ⁸
TransmissionSegment Object	i	i
ProfileValue Object	i	i
AccountRequest Object	i	i

Device Outage	Client	Server
DeviceOutage Object	O	O

InformationBuffer	Client	Server
InformationBuffer Object	O ⁴	O ⁴

Power Plant	Client	Server
AvailabilityReport Object	i	i
RealTimeStatus Object	i	i
ForecastSchedule Object	i	i
Curve Object	i	i
Power System Dynamic Objects	i	i

GeneralDataReport	Client	Server
GeneralDataReport Object	i	i
GeneralDataResponse Object	i	i

Annex A (informative)

TASE.2 (2002) Additional Object Models

A.1 General

IEC 60870-6-802:2002 included several normative object models that are out-of-scope in this version. The intent is to deprecate the use of these model elements in the next revision of IEC 60870-6-802. However, in order to provide a historical record, the information from the 2002 version is replicated in the following clauses.

A.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 following is a list of object models for transfer accounts.

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 <i>curtailment, halt, revision, or deletion</i> 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.

Data Type

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.

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 <u>only</u> 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 <u>only</u> 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.

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.

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.

A.3 Power Plant Objects**A.3.1 General**

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.

A.3.2 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.

A.3.3 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.

A.3.4 Forecast Schedule Object

A.3.4.1 General

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.

A.3.4.2 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 $A_0 + A_1x + A_2x^2 \dots A_nx^n$, 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.

A.3.4.3 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

A.4 General Data Report Object

A.4.1 General

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 IEC 60870-6-802, with the exception of a new text matrix data type and TransactionCode. Otherwise, only the abstract model and structure definitions differ from the standard.

A.4.2 General Data Request Object

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 <i>curtailment</i> , <i>halt</i> , <i>revision</i> , or <i>deletion</i> 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 Integer1 data 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 Text1 array.

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.

A.4.3 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.

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.

Annex B (informative)

TASE.2 (2002) Additional MMS Object Types

B.1 General

IEC 60870-6-802:2002 included several normative normative object types that are out-of-scope in this version. The intent is to deprecate the use of these types in the next revision of IEC 60870-6-802. However, in order to provide a historical record, the information from the 2002 version is replicated in Clauses B.2 to B.5.

B.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 }

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
}

```

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	Interchangeld,
COMPONENT InterchangePtOut	Interchangeld,
COMPONENT InterchangePt	Interchangeld,
COMPONENT NumberFloatIds	NumFloats,
COMPONENT NumberIntegerIds	NumIntegers

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	Interchangeld,	
COMPONENT InterchangePtOut	Interchangeld,	
COMPONENT InterchangePt	Interchangeld,	
COMPONENT NumberProfileValues	NumProfiles	
}		

ProfileValue Type Descriptions

TAProfileValue STRUCTURE

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

Account Request Type Descriptions

AccountRequest STRUCTURE

{		
COMPONENT ReferenceTar	ReferenceNum,	
COMPONENT StartTime	ScheduleTime,	
COMPONENT Duration	Duration,	
COMPONENT RequestId	ReferenceNum,	
COMPONENT TaConditionsRequested	TAConditions	
}		

B.3 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
}
```

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 bitstring { price[0], ramp[1], capacity[2]
},
```

COMPONENT PriceImpact	Price,
COMPONENT RampImpact	RampRates,
COMPONENT CapacityImpact	UnitCapacity,
COMPONENT Availability	AvailabilityClass,
COMPONENT Comment	VisibleString { width 255 }

}

UnAvailable STRUCTURE

{

COMPONENT AvailabilityReferenceld	ReferenceNum,
COMPONENT PlantReferenceld	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 }

}

Real Time Status Type Descriptions**StatusAvailable** STRUCTURE

{

COMPONENT AvailabilityReferenceld	ReferenceNum,
COMPONENT PlantReferenceld	PlantId,
COMPONENT UnitId	UnitId,
COMPONENT TimeStamp	TimeStampS,
COMPONENT RampStatus	RampRates,
COMPONENT CapacityStatus	UnitCapacity,
COMPONENT AvailabilityStatus	AvailabilityClass,
COMPONENT ProvidingReserve	boolean

}

StatusUnAvailable STRUCTURE

{

COMPONENT AvailabilityReferenceld	ReferenceNum,
COMPONENT PlantReferenceld	PlantId,
COMPONENT UnitId	UnitId,
COMPONENT TimeStamp	TimeStampS,
COMPONENT ReasonForUnavailable	AvailReasonCode,
COMPONENT ProvidingReserve	boolean

}

Forecast Type Descriptions**Forecast** STRUCTURE

{

COMPONENT ForecastScheduleReferenceld	ReferenceNum,
COMPONENT PlantReferenceld	PlantId,
COMPONENT UnitId	UnitId,

```

    COMPONENT StartTime
    COMPONENT PeriodResolution
    COMPONENT NumPeriods
    COMPONENT ForecastType
    GMTBasedS,
    TimeIntervals,
    integer,
    bitstring { generation[0], reserve[1] }
}

```

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
}

```

B.4 Power System Dynamics

B.4.1 General

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

B.4.2 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.

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.

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.

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

B.5 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
}
```

B.6 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
}
```

Annex C (informative)

TASE.2 (2002) Mapping of Objects to MMS Types

C.1 General

IEC 60870-6-802:2002 included several normative MMS types that are out-of-scope in this version. The intent is to deprecate the use of these types in the next revision of IEC 60870-6-802. However, in order to provide a historical record, the information from the 2002 version is replicated in the following clauses.

C.2 Transfer Accounts Mapping

C.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_Transmission_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 B.4.2 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 60870-6-503:2002);
- b) Request_Id variable, if the report is in response to a Query Operation (see 7.1.5.1.2 of IEC 60870-6-503:2002);
- 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	Data Type	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 C.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**.

Data Type

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.

C.2.2 TransmissionSegment Mapping

C.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.

C.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.

C.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**.

C.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**.

C.3 Power Plant Mapping

C.3.1 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.

PricelImpact

If present, maps to the **PricelImpact** COMPONENT of the MMS Named Variable of type **Available**. If bit 0 of the **Impact** COMPONENT is NO, the **PricelImpact** 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**).

C.3.2 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.

C.3.3 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**.

C.3.4 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.

C.4 General Data Report Mapping

C.4.1 General Data Request 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, ListOfInteger1Values, 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 60870-6-503:2002, 7.1.4.4)
- b) Request_Id variable, if the report is in response to a Query Operation (see 60870-6-503:2002, 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 **NumberOfText1COMPONENT** 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 **NumberOfText2COMPONENT** 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.

C.4.2 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**.

Annex D
(informative)

Transfer account examples

In this example scenario, utility A sells utility E 10 000 MW for the first hour and 20 000 MW for the second hour. The energy is transmitted to E via three different paths (through utility B, through utility C, and through utility D). There are thus six transmission segments:

Segment	First Hour (MW)	Second Hour (MW)
A-B	5 000	10 000
B-E	5 000	10 000
A-C	2 000	4 000
C-E	2 000	4 000
A-D	3 000	6 000
D-E	3 000	6 000

The total energy transferred is to be 10 000 MW for the first hour, and 20 000 MW for the second. For each segment, one floating point quantity is reported (e.g. MW). For the entire transaction, two floating point quantities are reported (e.g. price and total MW).

This transaction shall be represented as follows:

- a) The first MMS Named Variable describes the conditions which triggered the report:
 - name TA_Conditions_Detected, type TAConditionsDetected, value:
 - bitstring, with bit 0 set (BeforeTheHour)
- b) The next MMS Named variable is of type TASegmentsPeriodic, with name TA_Segments_Periodic, value:


```

            TA_Segments_Periodic
            {
                TransferAccountRef:      1 002
                SendUtility:              A
                RecvUtility:              E
                SellingUtility:           A
                BuyingUtility:            E
                TimeStamp:                <time of transmission>
                Name:                     <name>
                StartTime:                12:00
                PeriodResolution          3 600 s
                NumberLocalReferences:    0
                NumberOfSegments:        6
                NumberFloatIds           2
                NumberIntegerIds         0
                NumberPeriods             2
            }
            
```
- c) The next variable is of type TATransmissionSegment, name TA_Transmission_Segment, representing the header variable for the segment:

TA_Transmission_Segment

```

{
    TransmissionReference      (A-B)
    UtilWheeling:              B
    UtilPaying:                E
    TransmissionSegType:      OUT_ONLY
    UtilIn:                    A
    UtilOut:                   B
    InterchangePtIn:          <ignored>
    InterchangePtOut:         <ignored>
    InterchangePt:            <local reference of the tie>
    NumberFloatIds            1
    NumberIntegerIds         0
}

```

- d) The next variable reports the meaning of what is being reported for the segment. Since there is only one value being reported (one column in the conceptual matrix) there will be only one identifier:

Matrix_Id, type MatrixId, value: MW

- e) Since there are two time periods (hours) for the account report, there will be two variables (rows of the conceptual matrix) reported, each of length 1:

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 5 000)

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 10 000)

- f) The sequence (TA_Transmission_Segment, Matrix_Id, and Float_Array1) is repeated for each of the remaining segments being reported:

For segment 2:

```

TA_Transmission_Segment
{
    TransmissionReference      (B-E)
    ...
    NumberFloatIds            1
    NumberIntegerIds         0
}

```

Matrix_Id, type MatrixId, value: MW

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 5 000)

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 10 000)

For segment 3:

```

TA_Transmission_Segment
{
    TransmissionReference      (A-C)
    ...
    NumberFloatIds            1
    NumberIntegerIds         0
}

```

Matrix_Id MW

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 2 000)

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 4 000)

For segment 4:

```

TA_Transmission_Segment
{
    TransmissionReference      (C-E)
    ...
    NumberFloatIds            1
    NumberIntegerIds          0
}
Matrix_Id                    MW
    
```

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 2 000)

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 4 000)

For segment 5:

```

TA_Transmission_Segment
{
    TransmissionReference      (A-D)
    ...
    NumberFloatIds            1
    NumberIntegerIds          0
}
Matrix_Id                    MW
    
```

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 3 000)

Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 6 000)

For segment 6:

```

TA_Transmission_Segment
{
    TransmissionReference      (D-E)
    ...
    NumberFloatIds            1
    NumberIntegerIds          0
}
Matrix_Id                    MW
    
```

Name: Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 3 000)

Name: Float_Array1, type FloatArray1, value represents total MW (Float_Array1[0] = 6 000)

Finally, the variables which represent the overall transaction between A and E can be included:

```

Matrix_Id      PRICE
Matrix_Id      MW
    
```

Name: Float_Array1, type FloatArray1, values for price and total MW for the first hour:

Float_Array1[0] = 10 000, Float_Array1[1] = 50,00

Name: Float_Array1, type FloatArray1, values for price and total MW for the second hour:

Float_Array1[0] = 20 000, Float_Array1[1] = 50,00

SOMMAIRE

AVANT-PROPOS.....	96
INTRODUCTION.....	98
1 Domaine d'application	99
2 Références normatives	99
3 Termes et définitions	99
4 Abréviations	99
5 Modèles d'objets.....	100
5.1 Généralités	100
5.2 Système de commande, de surveillance et d'acquisition de données (SCADA).....	100
5.2.1 Généralités	100
5.2.2 Objet IndicationPoint (point de signalisation)	100
5.2.3 Objet ControlPoint (point de commande)	103
5.2.4 Modèle d'objet Protection Equipment Event (Evènementde l'équipement de protection)	105
5.3 Objet DeviceOutage (délestage d'appareil)	108
5.4 Objet InformationBuffer (tampon d'informations)	111
6 Types MMS pour l'échange d'objets.....	112
6.1 Généralités	112
6.2 Types de conduite de surveillance et d'acquisition de données	112
6.2.1 Descriptions du type IndicationPoint	112
6.2.2 Descriptions du type ControlPoint.....	115
6.2.3 Descriptions du type Protection Equipment.....	116
6.3 Descriptions du type Device Outage	117
6.4 Descriptions du type InformationBuffer	118
7 Correspondance des modèles d'objets avec les types MMS	119
7.1 Correspondance entre surveillance de conduite et données.....	119
7.1.1 Correspondance avec l'objet Indication.....	119
7.1.2 Correspondance avec l'objet ControlPoint	121
7.1.3 Correspondance avec Protection Event	122
7.2 Correspondance de Device Outage.....	125
7.3 Correspondance de Information Buffer	128
8 Utilisation des objets Supervisory Control (supervision).....	128
8.1 Généralités	128
8.2 Utilisation du modèle IndicationPoint	128
8.3 Utilisation du modèle ControlPoint	129
9 Conformité.....	130
Annexe A (informative) Modèles d'objets TASE.2 (2002) supplémentaires	131
A.1 Généralités	131
A.2 Comptes de Transfert	131
A.3 Objets PowerPlant (centrale électrique)	138
A.3.1 Généralités	138
A.3.2 Objet AvailabilityReport (compte rendu de disponibilité)	139
A.3.3 Objet Real Time Status (état en temps réel)	142
A.3.4 Objet Forecast Schedule (prévision de planification des révisions)	145

A.4	Objet General Data Report.....	148
A.4.1	Généralités	148
A.4.2	Objet General Data Request.....	148
A.4.3	Objet General Data Response	152
Annexe B (informative) Types d'objets MMS TASE.2 (2002) supplémentaires.....		154
B.1	Généralités	154
B.2	Types de Comptes de Transfert	154
B.3	Descriptions du type Power Plant.....	156
B.4	Power System Dynamics (dynamique du système de puissance)	159
B.4.1	Généralités	159
B.4.2	Type de données Matrix	159
B.5	Descriptions du type GeneralDataReport	161
B.6	Descriptions du type GeneralDataResponse	161
Annexe C (informative) Correspondance des objets TASE.2 (2002) avec les types MMS		163
C.1	Généralités	163
C.2	Correspondance avec les comptes de transfert.....	163
C.2.1	Correspondance avec TransferAccount	163
C.2.2	Correspondance de TransmissionSegment (segment de transmission)	168
C.2.3	Correspondance de ProfileValue	170
C.2.4	Correspondance de AccountRequest	171
C.3	Correspondance de Power Plant.....	171
C.3.1	Disponibilité de Report Mapping	171
C.3.2	Correspondance de Real Time Status.....	175
C.3.3	Correspondance de Forecast.....	177
C.3.4	Correspondance de Curve	178
C.4	Correspondance de General Data Report.....	179
C.4.1	Correspondance de General Data Request.....	179
C.4.2	Correspondance de General Data Response	183
Annexe D (informative) Exemples de comptes de transfert		185

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE –

Partie 6-802: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Modèles d'objets TASE.2

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 60870-6-802 a été établie par le comité d'études 57 de l'IEC: Gestion des systèmes de puissance et échanges d'informations associés.

Cette troisième édition annule et remplace la deuxième édition parue en 2002 et son amendement 1 (2005). Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- a) Les objets Accounts, Programs, Event Enrollment et Event Condition ont vu leur statut "normatif" modifié en "informatif". Il en découle une mise à jour des tableaux de conformité.

- b) Les services associés avec Accounts, Programs, Event Enrollment et Event Condition sont dorénavant hors du domaine d'application.
- c) Les blocs de conformité selon TASE.2 numérotés 6, 7, 8 et 9 ont été sortis du domaine d'application.
- d) Les mises en relation MMS pour les objets Accounts, Programs, Event Enrollment et Event Condition ont vu leur statut "normatif" modifié en "informatif".

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
57/1455/FDIS	57/1479/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 60870, publiées sous le titre général *Matériels et systèmes de téléconduite*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

INTRODUCTION

L'objectif principal de TASE.2 (Telecontrol Application Service Element = Élément de Service d'Application de Téléconduite) est de spécifier le transfert de données entre des systèmes de conduite et de lancer les actions de conduite. Les données sont représentées par des instances d'objets. La présente partie de la CEI 60870 propose des modèles d'objets à partir desquels on peut définir les instances d'objets. Les modèles d'objets représentent les objets à transférer. Le système local peut ne pas conserver de copie à jour de chaque attribut d'une instance d'objet.

Les modèles d'objets présentés ici sont spécifiques à l'opération et aux applications des «centres de conduite» ou des «compagnies de services publics»; on trouve les objets nécessaires à la mise en œuvre du protocole et des services TASE.2 dans la CEI 60870-6-503. Etant donné que les besoins évoluent, les modèles d'objets présentés ici ne fournissent qu'une base; des extensions ou des modèles supplémentaires peuvent être nécessaires pour que deux systèmes puissent échanger des données non définies dans la présente norme.

Par définition, les valeurs des attributs (c'est-à-dire les données) sont gérées par le propriétaire (c'est-à-dire l'émetteur) d'une instance d'objet. La méthode d'acquisition des valeurs dépend de la mise en œuvre; la précision est donc traitée localement.

La notation utilisée pour définir le modèle des objets spécifiés à l'Article 5 est définie dans l'IEC 60870-6-503. La présente partie de l'IEC 60870 est fondée sur les services et le protocole TASE.2. Afin de comprendre les procédures de création d'un modèle et la sémantique définies dans la présente norme, il est recommandé de se familiariser avec certaines notations de base de l'IEC 60870-6-503.

La notation utilisée pour définir le modèle des objets spécifiés en B.2 est définie dans l'IEC 60870-6-503. La présente partie de l'IEC 60870-6 est fondée sur les services et le protocole TASE.2. Afin de comprendre les procédures de création d'un modèle et la sémantique définies dans la présente partie de l'IEC 60870-6, il est recommandé de se familiariser avec certaines notations de base de l'IEC 60870-6-503.

L'Article 5 décrit les modèles d'objets spécifiques au centre de conduite et à leur application. Ils sont destinés à fournir des informations expliquant la fonction des données.

L'Article 6 définit un ensemble de descriptions de type MMS à utiliser lors de l'échange des valeurs des instances des modèles d'objets définis. Il est important de noter que tous les attributs des modèles d'objets ne correspondent pas à des types. Certains attributs ne sont décrits que pour définir le traitement demandé par le propriétaire des données et ne sont jamais échangés entre des centres de conduite. D'autres attributs servent à déterminer les types spécifiques des variables MMS utilisées pour la correspondance et n'apparaissent donc pas eux-mêmes comme valeurs échangées. Un modèle d'objet unique peut aussi correspondre à plusieurs variables MMS distinctes, selon le type d'accès et les services TASE.2 requis.

L'Article 7 décrit la correspondance entre les instances des variables MMS de chaque type d'objet et les listes de variables nommées pour la mise en œuvre d'échanges.

L'Article 8 décrit les codes et les sémantiques spécifiques aux dispositifs à employer pour les objets généraux.

L'Article 9 donne le tableau de conformité aux normes.

Une Annexe A informative a été ajoutée. Elle décrit des scénarios types de planification d'échanges, ainsi que l'utilisation des objets TASE.2 pour mettre en œuvre l'échange de planification.

MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE –

Partie 6-802: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Modèles d'objets TASE.2

1 Domaine d'application

La présente partie de l'IEC 60870 décrit une méthode d'échange de données «temps critique» de centres de conduite, à travers des réseaux locaux ou à longues distances utilisant une pile de protocoles entièrement conformes à l'ISO. Elle contient des dispositions permettant de soutenir aussi bien une architecture distribuée qu'une architecture centralisée. La norme traite d'échanges d'indications de données temps réel, d'opérations de conduite, de données par ordre chronologique, d'informations de planification et de comptabilité, de commande à distance du programme et de notification d'évènement.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60870-5-101:2003, *Matériels et systèmes de téléconduite – Partie 5-101: Protocoles de transmission – Norme d'accompagnement pour tâches élémentaires de téléconduite*

IEC 60870-6-503:2014, *Matériels et systèmes de téléconduite – Partie 6-503: Protocoles de téléconduite compatibles avec les normes ISO et les recommandations de l'UIT-T – Services et protocole TASE.2*

ISO 9506-1:2003, *Systèmes d'automatisation industrielle – Spécification de messagerie industrielle – Partie 1: Définition des services*

ISO 9506-2:2003, *Systèmes d'automatisation industrielle – Spécification de messagerie industrielle – Partie 2: Spécification de protocole*

3 Termes et définitions

Pour les besoins de la présente partie de l'IEC 60870, les termes et définitions des normes citées ci-dessus s'appliquent.

4 Abréviations

Pour les besoins de la présente partie de l'IEC 60870, toutes les abréviations définies dans les normes citées ci-dessus s'appliquent.

5 Modèles d'objets

5.1 Généralités

Dans un système, les modèles d'objets sont nécessaires à de diverses fonctions. L'Article 5 définit les modèles d'objets abstraits à partir de leurs fonctionnalités. Les modèles d'objets d'un domaine fonctionnel peuvent être utilisés dans un autre domaine fonctionnel.

5.2 Système de commande, de surveillance et d'acquisition de données (SCADA)

5.2.1 Généralités

Les modèles d'objets présentés dans cet article sont dérivés de la perspective historique des systèmes SCADA (Supervisory Control and Data Acquisition = systèmes de commande de surveillance et d'acquisition de données). Ce paragraphe présente le contexte dans lequel sont définis les modèles d'objets.

Fondamentalement, les systèmes SCADA ont deux fonctions clés: la commande et la signalisation. La fonction de commande est associée à l'émission de données tandis que la fonction de signalisation est associée à la réception de données. Un concept plus récent qui trouve son utilité est la fonction de commande et de signalisation, où les données peuvent être aussi bien émises que reçues (c'est-à-dire fonction bidirectionnelle).

Les fonctions identifiées précédemment dans les systèmes SCADA correspondent à des équipements de pointage (point). L'attribut principal d'un point est la valeur des données. Les systèmes SCADA définissent trois types de données pour les points: analogiques, numériques et d'état.

L'association d'un ou de plusieurs points sert à représenter des dispositifs. Par exemple, un dispositif de coupe-circuit peut être représenté par un point de commande et un point de signalisation. Le point de commande représente le nouvel état souhaité pour le dispositif de coupe-circuit. Le point de signalisation représente l'état actuel du dispositif de coupe-circuit. Pour les échanges de données de SCADA à SCADA (par exemple, de centre de conduite à centre de conduite, de centre de conduite à SCADA maître, etc.), on associe souvent des données supplémentaires aux données du point. La qualité des données du point est souvent transmise afin de définir si les données sont valides. En outre, pour les données qui peuvent être mises à jour à partir de sources de remplacement, la qualité identifie souvent la source de remplacement. Une commande Select-Before-Operate (Sélectionner-Avant-Opération) est associée aux Points de Commande pour empêcher momentanément l'accès à partir de toutes les sources, sauf une. Il existe deux autres valeurs de données informatives: l'horodateur et le changement de compteur de valeur. L'horodateur, quand il existe, précise la date de dernière modification de la valeur d'une donnée. Le changement de compteur de valeur, quand il existe, précise le nombre de modifications de la valeur.

A partir du contexte présenté, on déduit que les principaux modèles d'objets nécessaires sont: Indication Point (Point de Signalisation) et Control Point (Point de Commande). Les attributs Point Value (Valeur du Point), Quality (Qualité), Select-Before-Operate (Sélectionner-Avant-Opération), Time Stamp (Horodateur) et Change of Value Counter (Changement de Compteur de Valeur) sont nécessaires pour satisfaire aux fonctionnalités demandées pour les échanges de données. Les modèles Point de Signalisation et Point de Commande peuvent être combinés logiquement en un seul modèle représentant un dispositif mettant en œuvre une fonction de commande et une indication d'état concernant le succès ou l'échec. Le modèle logique combiné aura les mêmes attributs logiques et correspondra aux mêmes types MMS que les modèles indépendants.

5.2.2 Objet IndicationPoint (point de signalisation)

Un objet IndicationPoint représente un point d'entrée réel.

Object: **IndicationPoint** (Read Only)

Key Attribute: PointName
 Attribute: PointType (REAL, STATE, DISCRETE, STATESUPPLEMENTAL)
 Constraint PointType=REAL
 Attribute: PointRealValue
 Constraint PointType=STATE
 Attribute:PointStateValue
 Constraint PointType=DISCRETE
 Attribute: PointDiscreteValue
 Constraint PointType= STATESUPPLEMENTAL
 Attribute:PointStateSupplementalValue
 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 (horodateur)
 Attribute: TimeStampQuality: (VALID, INVALID)
 Constraint: TimeStampClass = TIMESTAMPEXTENDED
 Attribute: TimeStampExtended
 Attribute: TimeStampQuality: (VALID, INVALID)
 Attribute: COVClass: (COV, NOCOV)
 Constraint: COVClass = COV
 Attribute: COVCounter

PointName (nom du point)

L'attribut PointName sert à identifier l'objet de façon unique.

PointType (type de point)

L'attribut PointType identifie le type de point d'entrée et doit correspondre à l'un des types suivants: REAL (réel), STATE (état), DISCRETE (discret), STATESUPPLEMENTAL (état supplémentaire).

PointRealValue (valeur réelle du point)

La valeur courante de IndicationPoint, si l'attribut PointType est REAL.

PointStateValue (valeur d'état du point)

La valeur courante de IndicationPoint, si l'attribut PointType est STATE.

PointDiscreteValue (valeur discrète du point)

La valeur courante de IndicationPoint, si l'attribut PointType est DISCRETE.

PointStateSupplementalValue (valeur d'état supplémentaire du point)

La valeur courante de IndicationPoint, si l'attribut PointType est STATESUPPLEMENTAL.

L'attribut PointStateSupplementalValue doit pouvoir indiquer la valeur courante (State), les informations d'étiquetage (Tag), et la valeur/l'état attendu (ExpectedState). Si la valeur de ExpectedState ne correspond pas à la valeur State, ceci indique que le fournisseur de la valeur ExpectedState indique un problème potentiel.

QualityClass (classe de qualité)

QualityClass prend la valeur QUALITY si l'instance d'objet a l'un des attributs de qualité: Validity (validité), CurrentSource (émetteur actuel) ou NormalValue (valeur normale), et prend la valeur NOQUALITY si aucun de ces attributs n'est présent.

Validity (validité)

L'attribut Validity précise la validité ou la qualité des données de PointValue à laquelle elles sont associées. Leur interprétation est basée sur le système émetteur, comme suit:

Validity	Description
VALID	Valeur des données valide.
HELD	Valeur précédente des données conservée. Interprétation locale.
SUSPECT	Valeur des données douteuse. Interprétation locale.
NOTVALID	Valeur des données non valide.

CurrentSource (émetteur actuel)

L'attribut CurrentSource spécifie l'émetteur actuel des données de PointValue auquel elles sont associées, comme suit:

CurrentSource	Description
TELEMETERED	Valeur des données provenant d'un site de relevé à distance.
CALCULATED	Valeur des données calculée à partir d'autres valeurs de données.
ENTERED	Valeur des données entrée manuellement.
ESTIMATED	Valeur des données estimée. (State Estimator, etc.)

NormalSource (émetteur normal)

L'attribut NormalSource spécifie l'émetteur normal des données de PointValue auquel elles sont associées, comme suit:

NormalSource	Description
TELEMETERED	Valeur des données provenant normalement d'un site de relevé à distance.
CALCULATED	Valeur des données normalement calculée à partir d'autres valeurs de données.
ENTERED	Valeur des données normalement entrée manuellement.
ESTIMATED	Valeur des données normalement estimée. (State Estimator, etc.)

NormalValue (valeur normale)

L'attribut NormalValue indique si la valeur de l'attribut PointValue est normale. Il consiste en un seul bit défini comme suit:

NormalValue	Description
NORMAL	La valeur du point est celle qui a été configurée comme normale pour ce point.
ABNORMAL	La valeur du point n'est pas celle qui a été configurée comme normale pour ce point.

TimeStampClass (classe de l'horodateur)

L'attribut TimeStampClass prend la valeur `TIMESTAMP` ou `TIMESTAMPEXTENDED` si IndicationPoint peut fournir l'attribut TimeStamp, et la valeur `NOTIMESTAMP` dans le cas contraire.

TimeStamp (horodateur)

L'attribut TimeStamp donne la date et l'heure (avec une résolution minimale de une seconde) de dernière modification de la valeur (attribut PointRealValue, PointStateValue, PointDiscreteValue ou PointStateSupplementalValue) de IndicationPoint. On lui attribue la valeur dès que possible après la collecte des valeurs de IndicationPoint par le dispositif terminal.

TimeStampExtended (horodateur étendu)

L'attribut TimeStampExtended donne la date et l'heure (avec une résolution d'une milliseconde) de dernière modification de la valeur (attribut PointRealValue, PointStateValue, PointDiscreteValue ou PointStateSupplementalValue) de IndicationPoint. On lui attribue la valeur dès que possible après la collecte des valeurs de IndicationPoint par le dispositif terminal.

TimeStampQuality (qualité de l'horodateur)

L'attribut TimeStampQuality prend la valeur `VALID` si la valeur courante de l'attribut TimeStamp contient la date et l'heure de dernière modification de la valeur et prend la valeur `INVALID` dans tous les autres cas.

COVClass (classe de changements du compteur de valeur)

L'attribut COVClass (**C**hange **O**f **V**alue Counter = changement de compteur de valeur) prend la valeur `COV` si IndicationPoint contient un attribut COVCounter; sinon, il prend la valeur `NOCOV`.

COVCounter (compteur de changements de valeur)

L'attribut COVCounter indique le nombre de modifications de la valeur (attribut PointRealValue, PointStateValue, PointDiscreteValue ou PointStateSupplementalValue) de IndicationPoint. Il est incrémenté chaque fois que le propriétaire donne une nouvelle valeur à IndicationPoint.

5.2.3 Objet ControlPoint (point de commande)

L'objet ControlPoint fait entièrement partie des services fournis par TASE.2. Il sert à représenter des valeurs de données de types divers typiques de SCADA et des systèmes de gestion de l'énergie. Généralement, un objet ControlPoint sera associé à un objet donné du monde réel.

Object: **ControlPoint** (Write Only, sauf pour les attributs CheckBackName, Tag, State et 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 (nom du point de commande)

L'attribut ControlPointName sert à identifier l'objet de façon unique.

ControlPointType (type du point de commande)

La valeur de l'attribut ControlPointType pour une instance de ControlPoint sera COMMAND ou SETPOINT qui indique le type de dispositif contrôlé.

CommandValue (valeur de la commande)

L'attribut CommandValue indique la commande adressée à un dispositif.

SetPointType (type de la valeur de consigne)

La valeur de l'attribut SetPointType pour une instance de ControlPoint de ControlPointType SETPOINT sera REAL ou DISCRETE, qui indique le type de la valeur de consigne.

SetPointRealValue (valeur de consigne réelle)

L'attribut SetPointRealValue peut être défini avec la valeur en virgule flottante requise pour une valeur de consigne.

SetPointDiscreteValue (valeur de consigne entière)

L'attribut SetPointDiscreteValue peut être défini avec la valeur entière requise pour la valeur de consigne.

DeviceClass (classe de dispositif)

L'attribut DeviceClass d'une instance de ControlPoint prend la valeur SBO si le dispositif nécessite une opération "Select" (sélection) avant d'être exploité, et la valeur NONSBO dans le cas contraire.

CheckBackName (recontrôle du nom)

L'attribut CheckBackName contient une description symbolique de l'objet physique contrôlé. Ces données sont renvoyées par le système exploitant cet objet physique au système demandant l'opération, si bien que la personne ou le système demandant l'opération peut s'assurer que l'objet correct a été sélectionné.

State (état)

L'attribut State indique si ControlPoint est SELECTED (sélectionné) ou NOTSELECTED (non sélectionné).

Timeout (hors délai)

L'attribut Timeout d'une instance de ControlPoint prend la valeur maximale du délai autorisé pendant lequel ControlPoint de DeviceClass SBO peut rester SELECTED avant l'opération.

TagClass (classe de drapeau)

L'attribut TagClass d'une instance de ControlPoint prend la valeur TAGGABLE si l'instance contient un attribut Tag, la valeur NONTAGGABLE dans le cas contraire.

Tag (drapeau)

L'attribut Tag signale si ControlPoint est étiqueté ou non et, le cas échéant, le niveau d'étiquetage. L'attribut Tag peut prendre les valeurs NO-TAG, OPEN-AND-CLOSE-INHIBIT ou CLOSE-ONLY-INHIBIT.

Reason (raison)

L'attribut Reason contient un message qui indique la raison de l'étiquetage.

5.2.4 Modèle d'objet Protection Equipment Event (Evènement de l'équipement de protection)

Le modèle d'objet suivant représente les évènements générés par l'opération de l'équipement de protection. Les évènements initiaux sont générés par l'équipement de protection quand il détecte des défauts. Les **évènements** de déclenchement envoient les commandes aux circuits de sortie générés par l'équipement de protection lorsqu'il décide de déclencher le disjoncteur. Les deux évènements sont transitoires. Les modèles de évènement de protection sont basés sur l'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 (nom)

L'attribut Name sert à identifier l'évènement de protection de façon unique.

ElapsedTimeValidity (validité des intervalles de temps)

Les intervalles de temps (attribut EventDuration (durée de l'évènement), DurationTime (durée) ou OperatingTime (durée d'opération) en fonction du type d'évènement) sont valides s'ils sont correctement acquis. Si la fonction d'acquisition détecte des conditions non valides, l'attribut ElapsedTimeValidity prend la valeur INVALID. Il prend la valeur VALID dans le cas contraire.

Blocked (bloqué)

L'attribut Blocked prend la valeur BLOCKED si la valeur de l'évènement de protection est bloquée pour la transmission, et NOTBLOCKED dans le cas contraire. La valeur reste dans l'état qui était le sien avant d'être bloquée. Le blocage et le déblocage peuvent être lancés par un verrouillage local ou par une cause locale automatique.

Substituted (substitué)

L'attribut Substituted prend la valeur SUBSTITUTED si l'évènement est provoqué par l'action d'un opérateur (dispatcher) ou par une source automatisée.

Topical (actuel)

L'attribut Topical prend la valeur TOPICAL si la mise à jour la plus récente a été réussie, et NOTTOPICAL si la mise à jour n'a pas abouti dans un intervalle de temps donné ou si elle est indisponible.

EventValidity (validité d'un évènement)

L'attribut EventValidity prend la valeur INVALID si la fonction d'acquisition identifie des conditions anormales de la source d'information, et VALID dans les autres cas.

ProtectionClass (classe de protection)

L'attribut ProtectionClass identifie le type d'évènement de protection et prend l'une des valeurs suivantes: SINGLE ou PACKED.

EventState (état d'évènement)

L'attribut EventState d'un évènement de protection SINGLE prend la valeur de l'évènement de protection: OFF, ON ou INDETERMINATE.

EventDuration (durée de l'évènement)

L'attribut EventDuration prend la valeur de la durée de l'évènement (temps total pendant lequel le défaut était détecté) ou temps d'opération (intervalle de temps entre le début de l'opération et l'exécution de la commande de déclenchement).

EventTime (heure de l'évènement)

L'attribut EventTime donne l'heure du début de l'opération.

EventClass (classe de l'évènement)

L'attribut EventClass définit le type d'évènement de protection signalé. La valeur START indique le début d'un évènement, et TRIP un évènement de déclenchement.

StartGeneral (démarrage général)

La valeur NOSTART indique qu'il n'y a pas de démarrage général d'opération et START indique que l'évènement inclut un démarrage général de l'opération.

StartPhase1 (démarrage Phase 1)

La valeur NOSTART de StartPhase1 indique que la Phase L1 n'était pas impliquée dans l'évènement, START indique qu'elle l'est.

StartPhase2 (démarrage Phase 2)

La valeur NOSTART de StartPhase2 indique que la Phase L2 n'était pas impliquée dans l'évènement, START indique qu'elle l'est.

StartPhase3 (Démarrage Phase 3)

La valeur NOSTART de StartPhase3 indique que la Phase L3 n'était pas impliquée dans l'évènement, START indique qu'elle l'est.

StartEarth (démarrage terre)

La valeur NOSTART de StartEarth indique l'absence de détection de courant homopolaire, START indique qu'il y en a.

StartReverse (démarrage retour)

La valeur NOSTART de StartReverse indique l'absence de détection de directionnel amont, START en indique la présence.

DurationTime (durée)

Durée en millisecondes du début à la fin de l'opération.

StartTime (heure de début)

Heure de début d'opération de l'équipement de protection.

TripGeneral (déclenchement général)

L'attribut TripGeneral prend la valeur TRIP si une commande générale a été émise sur le circuit de sortie pendant l'opération, et la valeur NOTRIP dans le cas contraire.

TripPhase1 (déclenchement de la Phase1)

L'attribut TripPhase1 prend la valeur TRIP si une commande a été émise sur le circuit de sortie de la Phase L1 pendant l'opération, et la valeur NOTRIP dans le cas contraire.

TripPhase2 (déclenchement de la Phase 2)

L'attribut TripPhase2 prend la valeur TRIP si une commande a été émise sur le circuit de sortie de la Phase L2 pendant l'opération, et la valeur NOTRIP dans le cas contraire.

TripPhase3 (déclenchement de la Phase 3)

L'attribut TripPhase3 prend la valeur TRIP si une commande a été émise sur le circuit de sortie de la Phase L3 pendant l'opération, et la valeur NOTRIP dans le cas contraire.

OperatingTime (durée de l'opération)

Durée en millisecondes entre le début de l'opération et l'émission de la première commande sur le circuit de sortie.

TripTime (heure de déclenchement)

Heure de début de l'opération.

5.3 Objet DeviceOutage (délestage d'appareil)

Un objet DeviceOutage sert à communiquer des informations de planification concernant les délestages d'appareils. Il se compose d'un certain nombre d'objets qui définissent l'appareil qui sera (ou a été) touché et décrit la période de temps pendant laquelle surviendra le délestage.

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 (identificateur de référence du délestage)

L'attribut OutageReferenceId est une valeur de référence unique attribuée par l'émetteur pour identifier ce délestage particulier.

OwningUtilityID (identificateur du centre propriétaire)

ID du centre ou de la zone de contrôle possédant l'équipement.

TimeStamp (horodateur)

L'attribut TimeStamp indique l'heure à laquelle un objet DeviceOutage a été généré.

StationName (nom de la station)

Nom de la station où réside l'équipement, le circuit, etc. concerné.

DeviceType (type d'appareil)

Type de l'appareil attribué.

DeviceName (nom de l'appareil)

Nom de l'appareil attribué.

DeviceNumber (numéro de l'appareil)

L'attribut DeviceNumber fournit une qualification supplémentaire de DeviceName dans le cas où DeviceName pourrait ne pas fournir une spécificité suffisante. Par exemple, des lignes de transmission multiples peuvent connecter deux mêmes sous-stations et recevoir un DeviceName unique. Pour les identifier de façon unique, on attribue à chaque ligne un DeviceNumber.

DeviceRating (débit de l'appareil)

DeviceRating est donné en KV, MW, MVAR.

ActivityDateAndTime (date et heure de l'activité)

Heure à laquelle l'activité s'est produite ou, pour les activités planifiées ou annulées, heure à laquelle l'activité est prévue.

Activity (activité)

Activity décrit le type de l'activité objet du compte rendu. Un NEWPLAN (nouveau plan) ou un REVISE (plan révisé) exige des dates de début et de fin de délestage. Une CANCEL (annulation) ne demande qu'une date d'annulation, enregistrée dans ActivityDateAndTime. Une ACTUAL (activité en cours) demande la date à laquelle l'évènement s'est produit, enregistré dans ActivityDateAndTime, ainsi qu'une description de ce changement d'état.

PlanType (type de plan)

Un plan peut être fait d'activités planifiées (avec des dates formellement définies) ou de dates d'activités simplement estimées.

PlannedOpenOrOutOfServiceDateAndTime (date et heure planifiées de fin de service)

Date et heure auxquelles l'appareil sera mis hors service (ouverture des interrupteurs/coupe-circuits).

PlannedCloseOrInServiceDateAndTime (date et heure planifiées de reprise du service)

Date et heure auxquelles l'appareil sera remis en service (fermeture des interrupteurs/coupe-circuits).

OutagePeriod (périodicité des délestages)

OutagePeriod décrit la périodicité des délestages pour les plans à exécuter périodiquement, par exemple de façon hebdomadaire ou quotidienne.

OutageType (type de délestage)

OutageType décrit des raisons de mise hors service de l'équipement. FORCED (forcée) est un délestage contrôlé mais non prévu. MAINTENANCE (maintenance) est un délestage prévu à des fins de maintenance. PARTIAL (partielle) est un délestage au cours duquel seule une partie de la capacité est mise hors service. ECONOMY (économie) est un délestage planifié pour des raisons économiques. UNPLANNED (imprévue) est un délestage imprévu.

OutageAmountType (étendue du délestage)

OutageAmountType indique si le délestage est PARTIAL (partiel) (une certaine capacité est toujours disponible) ou FULL (complet) (aucune capacité n'est disponible).

Amount (quantité)

Pour les délestages partiels, indique la quantité de capacité normale qui sera indisponible pendant le délestage.

UpperOperatingLimit (limite supérieure d'opération)

Limite supérieure de la plage exploitable pendant le délestage.

LowerOperatingLimit (limite inférieure d'opération)

Limite inférieure de la plage exploitable pendant le délestage.

Class (classe)

Spécifie si l'appareil était en service ou hors service quand le délestage est survenu.

Action (action)

Décrit l'évènement ayant provoqué le délestage ou la remise en service.

AffectedAmount (quantité concernée)

Décrit la charge à porter lorsque le délestage s'est produit.

Comments (commentaires)

Champ de 128 caractères de texte ASCII qui peut être utilisé pour les commentaires de l'opérateur.

OutageEffect (conséquences du délestage)

Conséquences du délestage sur les capacités de transfert. Ce champ contient jusqu'à 128 caractères de texte ASCII et peut être utilisé pour décrire les effets du délestage sur les capacités de transfert.

5.4 **Objet InformationBuffer** (tampon d'informations)

L'objet InformationBuffer sert à envoyer des messages de plusieurs lignes de texte ASCII ou de données binaires. Il peut être utilisé pour transférer des données dont la taille est limitée par la taille maximale du message imposée par la structure de communication sous-jacente (c'est-à-dire la taille maximale d'une PDU MMS). L'utilisation et le codage du contenu de cet objet sont hors du domaine d'application de TASE.2. Ils sont traités localement et font l'objet d'accords entre les mises en œuvre émettrices et destinataires. A noter que cet objet est référencé comme l'objet Information Message dans l'IEC 60870-6-503.

Object: **InformationBuffer**

Key Attribute: InfoReference

Attribute: LocalReference

Attribute: MessageId

Attribute: Size

Attribute: InfoStream

InfoReference (Référence de l'information)

L'attribut InfoReference identifie l'objet de façon unique. Il sert à identifier et/ou à déclencher certains traitements spéciaux demandés par le système destinataire.

LocalReference (référence locale)

L'attribut LocalReference spécifie une valeur sur laquelle se sont accordés l'émetteur et le destinataire d'un message et qui va par la suite identifier le message (nom du fichier, identification de l'application, etc.).

MessageId (identificateur du message)

Identifie une instance particulière d'un message.

Size (taille)

Longueur des données valides du InfoStream.

InfoStream (flux d'informations)

L'attribut InfoStream contient le flux d'octets des informations transmises. Il n'est limité que par la taille maximale d'un message unique.

6 Types MMS pour l'échange d'objets**6.1 Généralités**

Cet article définit les Types MMS à utiliser dans TASE.2 pour échanger des objets normalisés. La correspondance entre les objets de ces types est définie à l'Article 7. Les définitions de type MMS sont faites en termes de notation de valeur ASN.1, selon la grammaire MMS utilisée pour les données, comme défini dans l'ISO 9506-1 et l'ISO 9506-2.

Tout au long de cet article, toutes les tailles des champs spécifiés correspondent aux tailles maximales des champs. Le procédé de codage ASN.1 utilisé dans MMS peut réduire les tailles réelles transmises au minimum nécessaire pour représenter la valeur transmise.

6.2 Types de conduite de surveillance et d'acquisition de données**6.2.1 Descriptions du type IndicationPoint**

Les types de base suivants sont référencés dans les descriptions du type IndicationPoint complexes:

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_StateSupplemental bit-string:

```
{
    State_hi[0],
    State_lo[1],
    tag_hi [2],
    tag_lo [3],
    ExpectedState_hi[4],
    ExpectedState_lo[5],
    Reserved[6],
    Reserved[7]
}
```

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 }

Les types complexes suivants sont utilisés pour transférer des valeurs d'objets IndicationPoint:

Data_RealQ STRUCTURE

```

{
    COMPONENT Value      Data_Real,
    COMPONENT Flags      Data_Flags
}

```

Data_StateQ STRUCTURE

```

{
    COMPONENT Value      Data_State,
    COMPONENT Flags      Data_Flags
}

```

Data_DiscreteQ STRUCTURE

```

{
    COMPONENT Value      Data_Discrete,
    COMPONENT Flags      Data_Flags
}

```

Data_StateSupplementalQ STRUCTURE

```

{
    COMPONENT Value      Data_StateSupplemental,
    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_StateSupplementalQTimeTag STRUCTURE
{
    COMPONENT Value      Data_StateSupplemental,
    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_StateSupplementalExtended STRUCTURE
{
    COMPONENT Value      Data_StateSupplemental,
    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
    }
Data_State_SupplementalQTimeTagExtended STRUCTURE
{
    COMPONENT Value              Data_StateSupplemental,
    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 }
}

```

6.2.2 Descriptions du type **ControlPoint**

Les types de base suivants sont référencés dans les descriptions de types complexes:

```

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 }

```

Les descriptions de types complexes suivantes sont utilisées pour accéder aux valeurs de l'objet **ControlPoint**:

```

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 }
}

```

}

6.2.3 Descriptions du type Protection Equipment

Les types de base suivants servent à bâtir des types complexes pour des évènements de l'équipement de protection du compte rendu:

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]
}
```

Les types complexes suivants sont utilisés pour des évènements de l'équipement de protection du compte rendu.

SingleProtectionEvent STRUCTURE

```
{
    COMPONENT SingleEventFlags           SingleFlags,
    COMPONENT OperatingTime              TimeInterval16,
    COMPONENT EventTime                   TimeStampExtended
}
```

PackedProtectionEvent STRUCTURE

```
{
    COMPONENT PackedEvent                 EventFlags,
    COMPONENT PackedEventFlags           PackedFlags,
    COMPONENT OperatingTime              TimeInterval16,
}
```

```

        COMPONENT EventTime                TimeStampExtended
    }

```

6.3 Descriptions du type Device Outage

Les types de base suivants sont référencés dans les descriptions de types complexes de l'objet DeviceOutage:

ActionId	integer {width 32}
ClassId	integer {width 16}
CommentString	VisibleString 128
Data_Real	floating-point {format-width 32, exponent-width 8}
Deviceld	integer {width 32}
DeviceName	visiblestring {width 32 characters, varying}
DeviceNumber	integer {width 32}
Number	integer {width 16}
OutageAmountTypeld	integer {width 16}
OutagePeriodId	integer {width 16}
OutageTypeld	integer {width 16}
PlanTypeld	integer {width 16}
ReferenceNum	integer {width 32}
ScheduleTime	GMTBasedS
StationNameString	VisibleString 32
TimeStampS	GMTBasedS
UtilityId	integer {width 32}

Les définitions de types composites de DeviceOutage sont:

DONewRevSched STRUCTURE

```

{
    COMPONENT OutageReferenceld                ReferenceNum,
    COMPONENT OwningUtilityId                 UtilityId,
    COMPONENT TimeStamp                       TimeStampS,
    COMPONENT StationName                     StationNameString,
    COMPONENT DeviceName                       DeviceName,
    COMPONENT DeviceType                       Deviceld,
    COMPONENT DeviceNumber                     Number,
    COMPONENT DeviceRating                     Data_Real,
    COMPONENT ActivityDateAndTime               ScheduleTime,
    COMPONENT PlanType                         PlanTypeld,
    COMPONENT PlannedOpenOrOutOfServiceDateAndTime ScheduleTime,
    COMPONENT PlannedCloseOrInServiceDateAndTime ScheduleTime,
    COMPONENT OutagePeriod                     OutagePeriodId,
    COMPONENT OutageType                       OutageTypeld,
    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 OutageReferenceld	ReferenceNum,
COMPONENT OwningUtilityId	UtilityId,
COMPONENT TimeStamp	TimeStampS,
COMPONENT StationName	StationNameString,
COMPONENT DeviceName	DeviceName,
COMPONENT DeviceType	Deviceld,
COMPONENT DeviceNumber	Number,
COMPONENT DeviceRating	Data_Real,
COMPONENT ActivityDateAndTime	ScheduleTime,
COMPONENT Comments	CommentString,
COMPONENT OutageEffect	CommentString

}

DOActual STRUCTURE

COMPONENT OutageReferenceld	ReferenceNum,
COMPONENT OwningUtilityId	UtilityId,
COMPONENT TimeStamp	TimeStampS,
COMPONENT StationName	StationNameString,
COMPONENT DeviceName	DeviceName,
COMPONENT DeviceType	Deviceld,
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 Descriptions du type InformationBuffer

Les types de base suivants sont référencés dans les descriptions de types complexes de l'objet InformationBuffer:

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

Les types suivants servent pour l'échange de types d'objets Message d'Informations:

InfoMessHeader STRUCTURE

COMPONENT InfoReference	ReferenceNum,
COMPONENT Localreference	ReferenceNum,
COMPONENT MessageId	ReferenceNum,
COMPONENT Size	Number

}

InfoBufXX OCTET STRING {width XX octets}

où XX est le nombre d'octets dans le tampon (buffer). Tous nombres d'octets sont autorisés, bien qu'il convient que la taille maximale du tampon ne dépasse pas la taille maximale de la PDU MMS.

Exemples: tampon de 64 octets = InfoBuf64, tampon de 256 octets = InfoBuf256, tampon de 1024 octets = InfoBuf1024. A noter que les zéros de gauche ne sont pas autorisés.

7 Correspondance des modèles d'objets avec les types MMS

7.1 Correspondance entre surveillance de conduite et données

7.1.1 Correspondance avec l'objet Indication

Le présent article définit la correspondance de chacun des attributs d'objet avec MMS. En général, la plupart des objets sont représentés par une ou plusieurs Variables Nommées MMS de types TASE.2 prédéfinis à l'Article 6.

PointName (nom du point)

Correspond à un identificateur de variable MMS (soit spécifique au VMD, soit spécifique au Domaine).

PointType (type du point)

Sert lors de la sélection du type nommé de la Variable. Si COVClass a la valeur NOCOV, le type de variable MMS est choisi en fonction des critères suivants:

PointType	QualityClass	TimeStampClass	Correspond au type:
REAL	NOQUALITY	NOTIMESTAMP	Data_Real
STATE	NOQUALITY	NOTIMESTAMP	Data_State
DISCRETE	NOQUALITY	NOTIMESTAMP	Data_Discrete
STATE SUPPLEMENTAL	NOQUALITY	NOTIMESTAMP	Data_StateSupplemental
REAL	QUALITY	NOTIMESTAMP	Data_RealQ
STATE	QUALITY	NOTIMESTAMP	Data_StateQ
DISCRETE	QUALITY	NOTIMESTAMP	Data_DiscreteQ
STATE SUPPLEMENTAL	QUALITY	NOTIMESTAMP	Data_StateSupplementalQ
REAL	QUALITY	TIMESTAMP	Data_RealQTimeTag
STATE	QUALITY	TIMESTAMP	Data_StateQTimeTag
DISCRETE	QUALITY	TIMESTAMP	Data_DiscreteQTimeTag
STATE SUPPLEMENTAL	QUALITY	TIMESTAMP	Data_StateSupplementalQTimeTag
REAL	QUALITY	TIMESTAMPEXTENDED	Data_RealQTimeTagExtended
STATE	QUALITY	TIMESTAMPEXTENDED	Data_StateQTimeTagExtended
DISCRETE	QUALITY	TIMESTAMPEXTENDED	Data_DiscreteQTimeTagExtended
STATE SUPPLEMENTAL	QUALITY	TIMESTAMPEXTENDED	Data_StateSupplementalQTimeTagExtended

Si COVClass a la valeur COV, les critères suivants sont utilisés:

PointType	Correspond au type:
REAL	Data_RealExtended
STATE	Data_StateExtended
DISCRETE	Data_DiscreteExtended
STATESUPPLEMENTAL	Data_StateSupplementalExtended

L'attribut PointType peut en option correspondre au composant **ControlPointType** d'une Variable Nommée MMS de type **IndicationPointConfig** avec la signification suivante: 0=STATE, 1=DISCRETE, 2=REAL, 3=StateSupplemental.

PointRealValue (valeur réelle du point)

Correspond, s'il est présent, soit à la valeur d'une variable MMS de type **Data_Real** (si QualityClass et TimeStampClass ont les valeurs NOQUALITY et NOTIMESTAMP), soit au composant **Value** de la variable MMS.

PointStateValue (valeur d'état du point)

Correspond, s'il est présent, soit à la valeur d'une variable MMS de type **Data_State** (si QualityClass et TimeStampClass ont les valeurs NOQUALITY et NOTIMESTAMP), soit aux bits **State_hi** et **State_lo** du composant **Flag** de la variable MMS.

PointDiscreteValue (valeur discrète du point)

Correspond, s'il est présent, soit à la valeur de la variable MMS de type **Data_Discrete** (si QualityClass et TimeStampClass ont les valeurs NOQUALITY et NOTIMESTAMP), soit au composant **Value** de la variable MMS.

PointStateSupplementalValue (valeur d'état supplémentaire du point)

Correspond, s'il est présent, soit à la valeur de la variable MMS de type **Data_StateSupplemental** (si QualityClass et TimeStampClass ont les valeurs NOQUALITY et NOTIMESTAMP), soit au composant **Value** de la variable MMS.

QualityClass (classe de qualité)

Sert au choix du type nommé de la variable (voir ci-dessus). L'attribut QualityClass peut aussi en option correspondre au composant **QualityClass** d'une Variable Nommée MMS de type **IndicationPointConfig** avec la signification suivante: NOQUALITY=0, QUALITY=1.

Validity (validité)

Correspond, s'il est présent, aux bits 2 et 3 (**Validity_hi**, **Validity_lo**) du composant **Flag** avec les valeurs suivantes: VALID = 0, HELD=1, SUSPECT=2, NOTVALID=3.

CurrentSource (émetteur actuel)

Correspond, s'il est présent aux bits 4 et 5 (**CurrentSource_hi**, **CurrentSource_lo**) du composant **Flags** avec les valeurs suivantes: TELEMETERED=0, CALCULATED=1, ENTERED=2, ESTIMATED=3.

NormalSource (émetteur normal)

L'attribut NormalSource peut en option correspondre au composant **NormalSource** d'une Variable Nommée MMS de type **IndicationPointConfig** avec la signification suivante: TELEMETERED=0, CALCULATED=1, ENTERED=2, ESTIMATED=3.

NormalValue (valeur normale)

Correspond, s'il est présent, au bit 6 (**NormalValue**) du composant **Flags** avec les valeurs suivantes: NORMAL=0, ABNORMAL=1.

TimeStampClass (classe de l'horodateur)

Sert au choix du type nommé de la variable (voir ci-dessus). L'attribut TimeStampClass peut aussi en option correspondre au composant **TimeStampClass** d'une Variable Nommée MMS

de type **IndicationPointConfig** avec la signification suivante: NOTIMESTAMP=0, TIMESTAMP=1, TIMESTAMPEXTENDED=2.

TimeStamp (horodateur)

Correspond, s'il est présent, au COMPOSANT **TimeStamp**.

TimeStampQuality (qualité de l'horodateur)

Correspond, s'il est présent, au bit 7 (**TimeStampQuality**) du composant **Flags** avec les valeurs suivantes: VALID=0, INVALID=1.

COVClass (classe COV)

Sert au choix du type nommé de la variable (voir ci-dessus). L'attribut COVClass peut aussi en option correspondre au composant **COVClass** d'une Variable Nommée MMS de type **IndicationPointConfig** avec la signification suivante: NOCOV=0, COV=1.

COVCounter (compteur COV)

Correspond, s'il est présent, à une variable MMS de type **COV_Counter**.

7.1.2 Correspondance avec l'objet ControlPoint

ControlPointName (nom du point de commande)

Correspond à un identificateur de variable MMS (soit spécifique au VMD, soit spécifique au Domaine).

ControlPointType (type du point de commande)

Sert au choix du type nommé de la Variable. Le type de la variable MMS est choisi selon les critères suivants (toute autre combinaison est non valide):

ControlPointType	SetPointType	Correspond au type:
COMMAND	Non applicable	Control_Command
SETPOINT	REAL	Control_Setpoint_Real
SETPOINT	DISCRETE	Control_Setpoint_Discrete

L'attribut ControlPointType peut aussi en option correspondre au composant **ControlPointType** d'une Variable Nommée MMS de type **ControlConfig** avec la signification suivante: COMMAND=0, SETPOINT=1.

CommandValue

Correspond à la valeur d'une variable MMS de type **Control_Command**.

SetPointType

Sert au choix du type nommé de la variable (voir ci-dessus). L'attribut Setpoint peut aussi en option correspondre au composant **SetpointType** d'une Variable Nommée MMS de type **ControlConfig** avec la signification suivante: 1=DISCRETE, 2=REAL.

SetpointRealValue

Correspond, s'il est présent, à la valeur d'une variable MMS de type **Control_Setpoint_Real**.

SetpointDiscreteValue

Correspond, s'il est présent, à la valeur d'une variable MMS de type **Control_Setpoint_Discrete**.

DeviceClass

L'attribut DeviceClass peut en option correspondre au composant **DeviceClass** d'une Variable Nommée MMS de type **ControlConfig** avec la signification suivante: 0=NONSBO, 1=SBO.

CheckBackName

Correspond, s'il est présent, à une variable MMS de type **SBO_CheckBackName**.

State

Correspond, s'il est présent, au composant **Select** d'une variable MMS de type **SBO** avec la signification suivante: SELECTED=TRUE, NOTSELECTED=FALSE.

Timeout

Correspond, s'il est présent, au composant **Timeout** d'une variable MMS de type **SBO**.

TagClass

L'attribut TagClass peut en option correspondre au composant **TagClass** d'une Variable Nommée MMS de type **ControlConfig** avec la signification suivante: 1=DISCRETE, 2=REAL.

Tag

Correspond, s'il est présent, aux bits 0 et 1 (**Tag_hi** et **Tag_lo**) du composant **Flags** d'une Variable Nommée MMS de type **Tag_Value** avec la signification suivante: NO-TAG=0, OPEN-AND-CLOSE-INHIBIT=1, CLOSE-ONLY-INHIBIT=2.

State

Correspond, s'il est présent, au bit 2 (**Tag_state**) du composant **Flags** d'une variable MMS de type **Tag_Value** avec la signification suivante: IDLE=0, ARMED=1.

Reason

Correspond, s'il est présent, au composant **Reason** d'une variable MMS de type **Tag_Value**.

7.1.3 Correspondance avec Protection Event

Ce paragraphe définit la correspondance de chaque attribut du modèle d'évènement de protection avec les types MMS. Les évènements de protection correspondent en général à des variables MMS. La méthode recommandée pour générer des comptes rendus d'évènement de protection est d'inclure les variables MMS dans un Data Set (jeu de données) TASE.2 et de renvoyer ce Data Set en utilisant un Data Set Transfer (transfert de jeu de données) en fixant les valeurs de DSTransmissionPars RBE à TRUE (Vrai) et de DSConditions y compris ObjectChange à TRUE (Vrai).

Name

L'attribut Name correspond à un identificateur de Variable Nommée MMS. A noter que Name sert à identifier de façon unique à la fois le dispositif de protection et le type d'évènement de protection.

ElapsedTimeValidity

L'attribut ElapsedTimeValidity correspond au bit 0 (**ElapsedTimeValidity**) du composant **SingleFlags** ou du composant **PackedFlags** de la variable MMS représentant l'évènement de protection, avec la valeur 0 représentant VALID et la valeur 1 représentant INVALID.

Blocked

L'attribut Blocked correspond au bit 1 (**Blocked**) du composant **SingleFlags** ou du composant **PackedFlags** de la variable MMS représentant l'évènement de protection, avec la valeur 0 représentant NOTBLOCKED et la valeur 1 représentant BLOCKED.

Substituted

L'attribut Substituted correspond au bit 2 (**Substituted**) du composant **SingleFlags** ou du composant **PackedFlags** de la variable MMS représentant l'évènement de protection, avec la valeur 0 représentant NOTSUBSTITUTED et la valeur 1 représentant SUBSTITUTED.

Topical

L'attribut Topical correspond au bit 3 (**Topical**) du composant **SingleFlags** ou du composant **PackedFlags** de la variable MMS représentant l'évènement de protection, avec la valeur 0 représentant TOPICAL et la valeur 1 représentant NOTTOPICAL.

EventValidity

L'attribut EventValidity correspond au bit 4 (**EventValidity**) du composant **SingleFlags** ou du composant **PackedFlags** de la variable MMS représentant l'évènement de protection, avec la valeur 0 représentant VALID et la valeur 1 représentant INVALID.

ProtectionClass

L'attribut ProtectionClass sert au choix du type de la variable MMS représentant l'évènement. La valeur de SINGLE implique l'utilisation d'une variable de type **SingleProtectionEvent** et la valeur PACKED implique l'utilisation d'une variable de type **PackedProtectionEvent**.

EventState

L'attribut EventState correspond aux bits 6 et 7 (**EventState_hi** et **EventState_lo**) du composant **SingleFlags** d'une variable MMS de type **SingleProtectionEvent**, avec 1 signifiant OFF, 2 signifiant ON et 0 ou 3 signifiant INVALID.

EventDuration

L'attribut EventDuration correspond au composant **OperatingTime** d'une variable MMS de type **SingleProtectionEvent**.

EventTime

L'attribut EventTime correspond au composant **EventTime** d'une variable MMS de type **SingleProtectionEvent**.

EventClass

L'attribut EventClass n'a pas de correspondant. La valeur de cet attribut est impliquée par l'identificateur de la variable MMS servant à représenter des événements de protection groupés; elle doit être connue et agréée par les parties intéressées.

StartGeneral

L'attribut StartGeneral correspond au bit 0 (**General**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

StartPhase1

L'attribut StartPhase1 correspond au bit 1 (**Phase1**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

StartPhase2

L'attribut StartPhase2 correspond au bit 2 (**Phase2**) du COMPOSANT **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

StartPhase3

L'attribut StartPhase3 correspond au bit 3 (**Phase3**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

StartEarth

L'attribut StartEarth correspond au bit 4 (**Earth**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

StartReverse

L'attribut StartReverse correspond au bit 5 (**Reverse**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOSTART et 1 représentant START.

DurationTime

L'attribut DurationTime correspond au composant **OperatingTime** d'une variable MMS de type **PackedProtectionEvent**.

StartTime

L'attribut StartTime correspond au composant **EventTime** d'une variable MMS de type **PackedProtectionEvent**.

TripGeneral

L'attribut TripGeneral correspond au bit 0 (**General**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOTRIP et 1 représentant TRIP.

TripPhase1

L'attribut TripPhase1 correspond au bit 1 (**Phase1**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOTRIP et 1 représentant TRIP.

TripPhase2

L'attribut TripPhase2 correspond au bit 2 (**Phase2**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOTRIP et 1 représentant TRIP.

TripPhase3

L'attribut TripPhase3 correspond au bit 3 (**Phase3**) du composant **EventFlags** d'une variable MMS de type **PackedProtectionEvent**, avec 0 représentant NOTRIP et 1 représentant TRIP.

OperatingTime

L'attribut OperatingTime correspond au composant **OperatingTime** d'une variable MMS de type **PackedProtectionEvent**.

TripTime

L'attribut TripTime correspond au composant **EventTime** d'une variable MMS de type **PackedProtectionEvent**.

7.2 Correspondance de Device Outage

Le présent article définit la correspondance du modèle d'objet Device Outage avec les types de MMS. Le modèle DeviceOutage correspond à une Variable Nommée MMS pour des transmissions utilisant le Compte rendu d'Informations MMS.

Les objets Device Outage correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chacune des Variables Nommées MMS utilisées pour la correspondance du modèle d'objet Device Outage n'est utilisée que dans les comptes rendus d'Informations MMS (MMS Information Reports) et renvoie un résultat d'accès MMS OBJECT-ACCESS-DENIED en lecture et en écriture.

Le type et le nom de la Variable Nommée MMS dépend de l'attribut Activity de l'évènement de Device Outage. La matrice suivante définit le choix du nom et du type:

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

OutageReferenceId

Correspond au composant **OutageReferenceId** du type de données sélectionné.

OwningUtilityID

Correspond au composant **OwningUtilityID** du type de données sélectionné.

Timestamp

Correspond au composant **TimeStamp** du type de données sélectionné.

StationName

Correspond au composant **StationName** du type de données sélectionné.

DeviceType

Correspond au composant **DeviveType** du type de données sélectionné, avec les significations suivantes: 1=GENERATOR, 2=TRANSFORMER, 3=CAPACITOR, 4=TRANSMISSION_CIRCUIT, 5=BREAKER_SWITCH, 6=INDUCTOR, 0=OTHER.

DeviceName

Correspond au composant **DeviceName** du type de données sélectionné.

DeviceNumber

Correspond au composant **DeviceNumber** du type de données sélectionné.

DeviceRating

Correspond au composant **DeviceRating** du type de données sélectionné.

ActivityDateAndTime

Correspond au composant **ActivityDateAndTime** du type de données sélectionné.

Activity

Sert au choix de la Variable Nommée MMS et du type utilisé pour la correspondance de l'objet.

PlanType

Correspond au composant **PlanType** d'une Variable Nommée MMS de type DONewRevSched, avec les significations suivantes: 0=SCHEDULED, 1=ESTIMATED.

PlannedOpenOrOutOfServiceDateAndTime

Correspond au composant **PlannedOpenOrOutOfServiceDateAndTime** d'une Variable Nommée MMS de type DONewRevSched.

PlannedCloseOrInServiceDateAndTime

Correspond au composant **PlannedCloseOrInServiceDateAndTime** d'une Variable Nommée MMS de type DONewRevSched.

OutagePeriod

Correspond au **OutagePeriod** d'une Variable Nommée MMS de type DONewRevSched, avec les significations suivantes: 1=CONTINUOUS, 2=DAILY, 3=WEEKDAYS, 0=OTHER.

OutageType

Correspond au **OutageType** d'une Variable Nommée MMS de type DONewRevSched, avec les significations suivantes: 0=FORCED, 1=MAINTENANCE, 2=PARTIAL, 3=ECONOMY, 4=UNPLANNED, 5=FORCED, 6=OTHER.

OutageAmountType

Correspond au **OutageAmountType** d'une Variable Nommée MMS de type DONewRevSched, avec les significations suivantes: 0=PARTIAL, 1=FULL.

Amount

S'il est présent, correspond au composant **Amount** d'une Variable Nommée MMS de type DONewRevSched.

UpperOperatingLimit

S'il est présent, correspond au composant **UpperOperatingLimit** d'une Variable Nommée MMS de type DONewRevSched.

LowerOperatingLimit

S'il est présent, correspond au composant **LowerOperatingLimit** d'une Variable Nommée MMS de type DONewRevSched.

Class

S'il est présent, correspond au composant **Class** d'une Variable Nommée MMS de type DONewRevSched avec les significations suivantes: 0=OUTSERVICE, 1=INSERVICE.

Action

S'il est présent, correspond au composant **Action** d'une Variable Nommée MMS de type DOActual avec les significations suivantes: 0=TRIPPED, 1=OFFLINE, 2=ONLINE, 3=OPEN, 4=CLOSE.

Affected Amount

S'il est présent, correspond au composant **AffectedAmount** d'une Variable Nommée MMS de type DOActual.

Comments

Correspond au composant **Comments** du type de données sélectionné.

OutageEffect

Correspond au composant **OutageEffect** du type de données sélectionné.

7.3 Correspondance de Information Buffer

L'objet Information Buffer correspond à deux Variables Nommées MMS. La première variable est de type **InfoMessHeader**. Elle contient les informations globales relatives au message. La seconde est de type **InfoBufXX**, où XX est assez grand pour contenir l'ensemble du message. La correspondance des attributs s'établit comme suit:

InfoReference

Correspond au COMPOSANT **InfoReference** d'une Variable Nommée MMS de type **InfoMessHeader**.

LocalReference

Correspond au COMPOSANT **LocalReference** d'une Variable Nommée MMS de type **InfoMessHeader**.

MessageId

Correspond au COMPOSANT **MessageId** d'une Variable Nommée MMS de type **InfoMessHeader**.

Size

Correspond au COMPOSANT **Size** d'une Variable Nommée MMS de type **InfoMessHeader**.

InfoStream

Correspond à une Variable Nommée MMS de type **InfoBufXX**, où XX est assez grand pour contenir la totalité des données. A noter que XX peut être plus grand, car l'attribut **Size** définit la part du tampon réellement valide.

8 Utilisation des objets Supervisory Control (supervision)

8.1 Généralités

Les modèles d'objets Supervisory Control (IndicationPoint et ControlPoint) sont génériques par nature, en ce sens qu'il est possible de représenter plusieurs types de dispositif avec ces objets de modèles. Le présent article fournit des exemples d'utilisation de ces modèles d'objets pour représenter des dispositifs réels. Il est cependant admis que cette liste peut ne pas être exhaustive. Si un nouveau dispositif est défini ultérieurement et nécessite une sémantique différente (c'est-à-dire, des interprétations) qui ne peut pas avoir de correspondance avec la liste existante, les implémenteurs peuvent alors ajouter une nouvelle sémantique tant qu'elle ne présente aucune contradiction avec la sémantique existante attribuée aux valeurs données dans l'Article 8.

8.2 Utilisation du modèle IndicationPoint

Le modèle IndicationPoint sert à représenter une saisie arbitraire de données à partir de dispositifs tels que des points d'état (PointType=STATE, PointType=STATESUPPLEMENTAL ou PointType=DISCRETE), des points analogiques (PointType=REAL) et des valeurs du compteur (PointType=DISCRETE) et des positions du Transformateur (PointType=DISCRETE).

PointType STATE et STATESUPPLEMENTAL sont recommandés pour les points d'état (simple ou double) contenant jusqu'à trois états, alors que PointType DISCRETE est recommandé pour les points d'état contenant plus de trois états. Les valeurs suivantes de

PointValue de type STATE sont utilisées pour représenter des positions spécifiques de dispositifs:

00	01	10	11	Device
Between	Tripped	Closed	Invalid	Disconnecter
Between	Off	On	Invalid	Disconnecter
Invalid	Off	On	Invalid	Breaker
Invalid	Auto	Manual	Invalid	
Invalid	Normal	Alarm	Invalid	
Invalid	Local	Remote	Invalid	
Invalid	Raise	Lower	Invalid	
Invalid	Not Ready	Ready	Invalid	
Invalid	Offline	AVAILABLE	Invalid	

Si DISCRETE est utilisé pour les informations des points simples et doubles, les valeurs suivantes de PointValue, de type DISCRETE (entier) sont utilisées pour représenter les positions spécifiques du dispositif:

0	1	2	3	Device
Between	Tripped	Closed	Invalid	Disconnecter
Between	Off	On	Invalid	Disconnecter
Invalid	Off	On	Invalid	Breaker
Invalid	Auto	Manual	Invalid	
Invalid	Normal	Alarm	Invalid	
Invalid	Local	Remote	Invalid	
Invalid	Raise	Lower	Invalid	
Invalid	Not Ready	Ready	Invalid	
Invalid	Offline	Available	Invalid	

Des saisies (de données) analogiques peuvent être représentées soit par PointType=REAL (si l'échelle et les procédures de normalisation sont appliquées du côté du serveur TASE.2), soit par PointType=DISCRETE (si on saisit les valeurs brutes).

8.3 Utilisation du modèle ControlPoint

Le modèle ControlPoint sert à représenter une saisie arbitraire de données sous forme de commandes de commutation vers des dispositifs tels que transformateurs ou dispositifs de commutation (ControlPointType=COMMAND), avec des valeurs de consigne analogiques ou numériques vers des dispositifs ou des unités tels que des unités de puissance (ControlPointType = SETPOINT). Les valeurs suivantes, de type COMMAND (entier), sont utilisées pour représenter des commandes de commutation du dispositif:

0	1	Device
Trip	Close	Switch
Open	Close	Switch
Off	On	Switch
Lower	Raise	Transformer

Les saisies (de données) analogiques peuvent être modélisées, soit par ControlPointType=REAL (si l'échelle et les procédures de normalisation sont appliquées du côté du serveur TASE.2), soit par ControlPointType=DISCRETE (si on spécifie des valeurs brutes).

9 Conformité

Les modèles d'objet de la présente partie de l'IEC 60870 ont été regroupés conformément aux blocs de conformité des services, comme défini à l'Article 9 de l'IEC 60870-6-503:2014. Les tableaux suivants définissent en détails les exigences de conformité des mises en œuvre de TASE.2. Dans ces tableaux, l'entrée O implique une option et Mⁿ implique que la construction est obligatoire pour le bloc de conformité n, comme défini ci-dessous. "i" indique que les objets ne relèvent pas du domaine d'application (hors portée) de la partie normative du présent document.

Système de commande, de surveillance et d'acquisition de données (SCADA)

	Client	Serveur
Objet IndicationPoint	M ¹	M ¹
Objet IndicationPoint (STATESUPPLEMENTAL)	O ¹	O ¹
Objet ControlPoint	M ⁵	M ⁵
Objet ProtectionEquipmentEvent	O	O

Comptes de Transfert

	Client	Serveur
Objet TransferAccount	i ⁸	i ⁸
Objet TransmissionSegment	i	i
Objet ProfileValue	i	i
Objet AccountRequest	i	i

Device Outage

	Client	Serveur
Objet DeviceOutage	O	O

InformationBuffer

	Client	Serveur
Objet InformationBuffer	O ⁴	O ⁴

Centrale électrique

	Client	Serveur
Objet AvailabilityReport	i	i
Objet RealTimeStatus	i	i
Objet ForecastSchedule	i	i
Objet Curve	i	i
Objets Power System Dynamic	i	i

GeneralDataReport

	Client	Serveur
Objet GeneralDataReport	i	i
Objet GeneralDataResponse	i	i

Annexe A (informative)

Modèles d'objets TASE.2 (2002) supplémentaires

A.1 Généralités

L'IEC 60870-6-802:2002 comprenait plusieurs modèles d'objets normatifs qui ne relèvent pas du domaine d'application de la présente version. L'intention est de déconseiller l'utilisation de ces éléments de modèles dans la prochaine révision de l'IEC 60870-6-802. Cependant, à titre historique, les informations de la version 2002 sont reproduites dans les articles suivants.

A.2 Comptes de Transfert

L'une des exigences clé des applications des centres de conduite est l'aptitude à échanger des informations de «programmation» et de «comptabilité». Dans le monde des centres de distribution, «programmes» signifie généralement une quantité d'énergie électrique transférée d'un système à un autre, sur une base périodique, pendant un intervalle de temps donné, sous réserve d'un accord formel. Concernant l'échange de données, le sens du terme échange de «programmes» a été étendu et comprend l'échange de toute donnée périodique ou de profil pour les applications de programmation, de comptabilité et de surveillance de l'énergie des centres de conduite. L'échange des programmes peut donc également signifier l'échange de données de production, échange réel, de charges, d'informations sur la tarification, de comptes rendus, etc. Répétons que toute information peut être échangée si elle peut être modélisée sous la forme de données périodiques ou de profil.

Les comptes de programme et les comptes d'informations comptables détaillent une liste de grandeurs (énergie, capacité ou toute autre grandeur définie par l'utilisateur) à échanger pendant des séquences temporelles. Ces transactions peuvent être détaillées soit comme valeurs périodiques à plat soit comme des profils en rampe. Les comptes de programme peuvent en outre contenir une liste de segments de transmission définissant les chemins utilisés pour mettre en œuvre la transaction.

La liste suivante donne les modèles d'objets pour les comptes de transfert.

Objet TransferAccount (compte de transfert)

Un objet TransferAccount représente l'objet, le lieu, la date et la quantité transférés entre deux sociétés de services d'un compte particulier. Il peut aussi représenter, dans une société de services, les prévisions de production ainsi que les autres planifications de livraison d'énergie. Il contient un certain nombre d'attributs et d'objets différents qui, ensemble, définissent entièrement le compte de transfert, c'est-à-dire quel compte, quelle est sa période de validité et quelle est la valeur des données horaires ou des profils de données. Pour plus d'informations sur la relation entre les différents objets, voir l'Annexe A.

Object: **TransferAccount**

Key Attribute: TransferAccountReference

Attribute: SendUtility

Attribute: ReceiveUtility

Attribute: SellingAgent

Attribute: BuyingAgent

Attribute: TimeStamp (horodateur)

Attribute: TransactionCode

Attribute: NumberOfLocalReference

Attribute: ListOfLocalReference

Attribute: Name
 Attribute: TransmissionSegmentOption (INCLUDED, NOTINCLUDED)
 Constraint: TransmissionSegmentOption=INCLUDED
 Attribute: NumberOfTransSegments
 Attribute: ListOfTransmissionSegments
 Attribute: DataType (PERIODIC, PROFILE)
 Constraint: DataType = PERIODIC
 Attribute: StartTime
 Attribute: PeriodResolution
 Attribute: NumberOfPeriods
 Attribute: ListOfPeriodicValues
 Constraint: DataType = PROFILE
 Attribute: NumberOfProfiles
 Attribute: ListOfProfileValues

TransferAccountReference (référence du compte de transfert)

L'attribut TransferAccountReference spécifie une valeur de référence unique entre l'émetteur et le destinataire pour identifier ce compte de transfert particulier.

SendUtility (centre de distribution émetteur)

L'attribut SendUtility spécifie la compagnie délivrant initialement l'énergie ou le service.

ReceiveUtility (centre de distribution destinataire)

L'attribut ReceiveUtility spécifie la compagnie recevant in-fine l'énergie ou le service.

SellingAgent (agent vendeur)

Pour le moment, l'agent vendeur est presque toujours confondu avec la SendUtility. Comme conséquence de la dérégulation et de l'ouverture des accès, il est cependant probable que, dans l'avenir, le vendeur puisse être un producteur d'énergie indépendant de la zone desservie par la SendUtility. Dans ce cas il est évidemment nécessaire de faire une distinction entre le centre vendeur et le centre émetteur. Le présent attribut fait cette distinction.

BuyingAgent (agent acheteur)

Pour le moment, l'agent acheteur est presque toujours confondu avec la ReceiveUtility. Comme conséquence de la dérégulation et de l'ouverture des accès, il est cependant probable que, dans l'avenir, l'acheteur puisse être une municipalité ou une coopérative de la zone desservie par la ReceiveUtility. Dans ce cas il est évidemment nécessaire de faire une distinction entre le centre acheteur et le centre destinataire. Le présent attribut fait cette distinction.

TimeStamp (horodateur)

L'attribut TimeStamp permet au destinataire de connaître la date et l'heure de l'envoi du message par l'émetteur. L'horodateur indique l'heure à laquelle l'application a créé les données du Compte de Transfert en vue de les transmettre. Il ne concerne pas l'heure à laquelle s'appliquent les données du Compte de Transfert. Si les applications qui fournissent les données à TASE.2 comportent la date et l'heure de la transmission (ou la création d'un objet), cette date et cette heure sont alors utilisées. Sinon, TASE.2 fournit la date et l'heure avant de soumettre l'objet Transfer Account à MMS.

TransactionCode (code de transaction)

L'attribut TransactionCode indique l'état de la transaction TransferAccount. Les valeurs sont données ci-dessous:

TransactionCode	Description
NEW	Soumission initiale d'une nouvelle proposition de programme ou de compte rendu
REQUESTED	Demande du compte rendu ou du programme via l'opération Account Request
REVISED	Version révisée du programme ou du compte rendu précédemment transmis
CONFIRMED	Le programme ou le compte rendu est à présent planifié pour mise en œuvre réelle sans nécessiter d'autres cycles d'approbation. Une opération de <i>curtailment</i> (<i>compression</i>), <i>halt</i> (<i>arrêt</i>), <i>revision</i> (<i>révision</i>) ou <i>deletion</i> (<i>annulation</i>) après cette étape nécessite de transmettre un nouveau programme ou compte rendu
DELETED	Tout moment auquel l'émetteur du programme ou du compte rendu annule la transaction (par exemple, du fait d'une panne de l'équipement)

NumberOfLocalReference (nombre de références locales)

Cet attribut contient le nombre d'entrées de la ListOfLocalReference.

ListOfLocalReference (liste de références locales)

Cette liste contient les paramètres définis localement qui spécifient comment l'objet Compte de Transfert est à traiter comme un ensemble. Par exemple, on peut spécifier que, pendant l'été, on planifie d'effectuer les transactions tous les lundis. La signification de la liste des paramètres est traitée localement et fait l'objet d'un accord entre les parties intéressées pour chaque type d'objet Compte de Transfert, pour tous les objets Compte de Transfert ou pour une combinaison des deux.

Name (nom)

L'attribut Name fournit une option à l'application utilisateur, pour inclure des chaînes d'informations associées à l'objet Transfer Account. Par exemple, un programme peut avoir besoin d'identifier une unité de production par son nom. Que cet attribut soit ou non utilisé, sa sémantique est un problème de mise en œuvre locale.

TransmissionSegmentOption (option du segment de transmission)

L'attribut TransmissionSegmentOption indique si on a ou non inclus une ListOfTransmissionSegment dans l'objet Transfer Account. Les segments de transmission peuvent être inclus dans l'objet Transfer Account indépendamment du type de données que l'objet Transfer Account représente, c'est-à-dire qu'il n'est pas nécessaire de les inclure uniquement pour des transactions de vente ou d'achat d'énergie entre compagnies quand le transit est impliqué. Les segments de transmission peuvent être utilisés dans tout objet Transfer Account pour véhiculer toute information entre plusieurs parties quand une partie de l'information est propre à chaque partie et que l'on souhaite fournir toutes les informations du compte à toutes les parties.

NumberOfTransSegments (nombre de segments de transmission)

Cet attribut définit le nombre de segments de transmission à inclure dans le compte rendu.

ListOfTransmissionSegments (liste de segments de transmission)

L'attribut ListOfTransmissionSegments décrit les diverses voies (ou segments) de transmission que peut prendre un échange de planification pour aller d'un SendUtility à un

ReceiveUtility, ou pour décrire d'autres informations significatives pour les parties intermédiaires d'une transaction. Chaque objet TransmissionSegment fourni décrit un élément de cette voie ou les données d'une partie intermédiaire. On peut mettre autant (ou aussi peu) d'objets TransmissionSegment qu'on le désire dans la liste.

DataType (type de données)

Les données contenues dans l'objet Transfer Account peuvent être périodiques; c'est le cas de la plupart des données utilisées pour la planification de production, les transactions d'énergie et la facturation. Les données peuvent aussi définir une série d'exigences croissantes qui aboutissent ensemble à un profil d'énergie (ou de tarif). L'attribut DataType indique lequel de ces deux types de données est contenu dans l'objet Transfer Account. A noter qu'un objet Transfer Account ne peut pas contenir à la fois des données périodiques et des données de profil. Il n'y a cependant pas de limite d'utilisation de l'attribut TransferAccountReference pour le type de données des objets. On peut en conséquence transmettre deux objets Transfer Account avec le même TransferAccountReference, le premier véhiculant des informations de type périodique et le second des données de type profil.

StartTime (heure de début)

Pour un objet Transfer Account contenant des données périodiques, StartTime spécifie l'heure UTC à laquelle s'applique la première période de la séquence.

PeriodResolution (durée d'une période)

Pour un objet Transfer Account contenant des données périodiques, PeriodResolution spécifie la durée pendant laquelle s'applique chaque entrée de la séquence.

NumberOfPeriods (nombre de périodes)

Cet attribut définit le nombre de périodes horaires spécifiées dans la séquence définie par l'objet Transfer Account. Avec StartTime et PeriodResolution, cet attribut définit aussi la durée maximale de temps couvert par l'objet Transfer Account.

ListOfPeriodicValues (liste de valeurs périodiques)

Cet attribut contient la masse des données associées au compte de transfert. Pour des transactions d'énergie, il s'agit des données relatives à l'utilisateur final de la transaction, c'est-à-dire la transaction entre l'acheteur et le vendeur. Les informations spécifiques aux partenaires du transit pour chaque segment de transmission sont spécifiées dans l'attribut ListOfTransmissionSegment (quand il est inclus).

NumberOfProfiles (nombre de profils)

Cet attribut définit le nombre de périodes de temps spécifiées dans ListOfProfileValues.

ListOfProfileValues (liste de valeurs de profil)

L'attribut ListOfProfileValue décrit le profil ou les informations croissantes. Le nombre d'objets ProfileValue transmis dans le message implique le nombre de modifications des points d'inflexion dans le profil, un pour chaque modification.

Objet TransmissionSegment (segment de transmission)

Un objet TransmissionSegment sert à représenter les spécificités d'un accord de transit dans lequel une compagnie autorise une autre compagnie à utiliser son réseau haute tension pour

transférer de l'énergie à une troisième compagnie. Cet objet peut aussi servir à véhiculer des informations vers une compagnie impliquée dans une transaction à plusieurs compagnies. Le `TransmissionSegment` peut décrire soit l'endroit où l'énergie est à recevoir et de quelle compagnie elle provient (INONLY), soit l'endroit où l'énergie est à livrer et quelle compagnie doit la recevoir (OUTONLY). Un `TransmissionSegment` peut également décrire les compagnies émettrices et réceptrices ainsi que les reçus et points de livraison associés (INOUT). Ou, si l'échange d'énergie est prévu directement entre deux compagnies (DIRECT), il peut ne spécifier que le point d'échange. Il est également possible de spécifier la compagnie prestataire et la compagnie payant la transmission (transit).

Object: TransmissionSegmentAttribute: `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** (référence de la transmission)

L'attribut `TransmissionReference` spécifie une valeur ayant fait l'objet d'un accord entre l'émetteur et le destinataire du message qui décrira ensuite le segment de transmission (numéro de référence de l'accord de transmission, etc.).

UtilWheeling (centre transporteur)

L'attribut `UtilWheeling` spécifie la compagnie fournissant les services de transmission (le cas échéant).

TransmissionSegType (type de segment de transmission)

L'attribut `TransmissionSegType` spécifie comme suit le type de données de l'objet `TransmissionSegment`:

TransmissionSegType	Description
INONLY	Cet objet <u>ne spécifie que</u> la compagnie d'où provient l'énergie et le lieu où elle est à recevoir. Ce type d'objet doit être utilisé quand le point de sortie du segment est inconnu ou sans importance.
OUTONLY	Cet objet <u>ne spécifie que</u> la compagnie où l'énergie est à livrer et le lieu où il faut la livrer. Ce type d'objet doit être utilisé quand le point d'entrée du segment est inconnu ou sans importance.
INOUT	Cet objet spécifie à la fois les compagnies réceptrices et émettrices ainsi que les points de réception et de livraison associés.
DIRECT	Cet objet ne spécifie que le point d'échange entre la compagnie origine envoyant l'énergie et la compagnie qui la reçoit in-fine. Ce type d'objet doit être utilisé quand aucun centre intermédiaire ne fournit les services de transmission.

UtilIn (compagnie fournisseur)

L'attribut UtilIn spécifie la compagnie d'où provient l'énergie importée.

InterchangePtIn (point d'échange en entrée)

L'attribut InterchangePtIn spécifie le point d'échange ou l'interface où l'énergie importée de UtilIn est à recevoir.

UtilOut (compagnie destinataire)

L'attribut UtilOut spécifie la compagnie auquel on livre l'énergie exportée.

InterchangePtOut (point d'échange en sortie)

L'attribut InterchangePtOut spécifie le point d'échange ou l'interface où on livre l'énergie exportée vers UtilOut.

InterchangePt (point d'échange)

L'attribut InterchangePt spécifie le point d'échange ou l'interface où s'effectue le transfert direct d'énergie entre SendUtility et RecvUtility.

UtilPaying (compagnie payeur)

L'attribut UtilPaying spécifie la compagnie auquel il convient que les frais pour tout service de transmission (transit, pertes, etc.) soient facturés.

ListOfSegmentData (liste de segments de données)

ListOfSegmentData contient les informations spécifiques aux partenaires de transit de chaque segment de transmission. Il peut contenir des données telles que l'énergie transférée au travers du segment et/ou le coût.

Objet ProfileValue (valeur de profil)

Un objet ProfileValue représente un profil dans le temps d'une grandeur ou valeur échangée. La valeur peut être de l'énergie, une capacité, un tarif et/ou toute autre information. Il est représenté en précisant une valeur cible, une heure de début de croissance et un taux de croissance. Quand on reçoit un objet ProfileValue qui affecte la grandeur de l'échange, le profil évolue dans la direction de la nouvelle cible, en commençant à l'heure définie par RampStartTime et au taux défini par RampDuration. Si RampDuration a la valeur zéro (ou s'il

n'est pas transmis), la valeur ne commence pas à progresser mais est fournie instantanément à RampStartTime. Lorsque la valeur cible est atteinte, la progression continue selon un taux constant. Ce taux constant peut être modifié ou supprimé à l'aide d'un autre objet ProfileValue.

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 (heure de début de croissance)

Heure à laquelle le changement est prévu de survenir.

RampDuration (durée de la croissance)

Temps pendant lequel la croissance peut s'effectuer. Les unités sont spécifiées dans des accords bilatéraux.

ProfilePrice (profil de prix)

La valeur du prix est généralement exprimée en unités monétaires par unité de mesure (telle que \$/MWH), telle qu'elle est définie dans les accords bilatéraux.

TargetClass (classe cible)

L'attribut TargetClass détermine le type de profil. Il prend les valeurs ENERGY, CAPACITY ou OTHER.

ProfileEnergy (énergie de profil)

La quantité d'énergie est généralement exprimée en quantité d'énergie/heure (telle que MWH) telle qu'elle est définie dans les accords bilatéraux. Cet attribut existe si l'attribut TargetClass est ENERGY.

ProfileCapacity (capacité de profil)

La valeur de la capacité est généralement exprimée comme une quantité de capacité (telle que MW) telle qu'elle est définie dans les accords bilatéraux. Cet attribut existe si l'attribut TargetClass est CAPACITY.

ProfileOther (autre profil)

Autre valeur non spécifiée, telle qu'elle est définie dans les accords bilatéraux. Utiliser l'interprétation locale. Cet attribut existe si la valeur de l'attribut TargetClass est OTHER.

Objet AccountRequest (demande de comptes)

Un objet AccountRequest sert à demander des informations sur les comptes. Cette requête peut être utilisée pour des comptes de planification d'échanges et/ou des comptes d'information de comptabilité. Cet objet, quand il a été écrit, provoque la génération par le serveur TASE.2 d'un Transfer Account Transfer Report (Compte rendu de Transfert de Compte de Transfert) TASE.2 (voir l'IEC 60870-6-503) des données requises. Le Rapport de Transfert contient l'identificateur AccountRequest en tant que variable MMS à transmettre.

Object: AccountRequest

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

AccountRequestName (nom de la demande de compte)

AccountRequestName sert à identifier l'objet AccountRequest de façon unique.

TransferAccountReference (référence du compte de transfert)

L'attribut TransferAccountReference spécifie une valeur de référence unique entre l'émetteur et le destinataire pour identifier ce compte de transfert particulier.

StartTime (heure de début)

L'attribut StartTime donne l'heure de démarrage demandée. Cette heure est spécifiée en secondes.

Duration (durée)

L'attribut Duration spécifie l'heure de fin demandée. Il donne le nombre de secondes à partir de StartHour.

RequestId (identificateur de la demande)

L'attribut RequestId sert à faire correspondre une réponse avec une demande. Cette valeur est renvoyée dans le compte rendu du Compte de Transfert.

TaConditionsRequested (conditions demandées pour la transaction)

L'attribut TaConditionsRequested identifie le type de données demandées.

A.3 Objets PowerPlant (centrale électrique)

A.3.1 Généralités

Ces objets sont prévus pour être utilisés avec les services TASE.2, tels qu'ils sont définis dans l'IEC 60870-6-503. On doit rendre compte des deux premiers objets en utilisant les mêmes mécanismes que pour le modèle d'objet de planification ou compte de TASE.2, avec sans doute des codes **Condition** redéfinis. Le troisième objet peut nécessiter un mécanisme légèrement différent à ajouter à TASE.2 pour permettre une négociation simple.

A.3.2 Objet AvailabilityReport (compte rendu de disponibilité)

L'objet suivant représente un compte rendu d'une centrale électrique, soit à un centre de conduite des unités de production (GCS), soit à un centre de conduite EMS annonçant la disponibilité globale d'une unité pour une période donnée à venir. Ce compte rendu est basé sur la disponibilité globale prévue des ressources de la centrale électrique. L'unité peut être capable de fonctionner avec des modes d'opération variés, chacun d'eux pouvant impliquer une structure différente de tarification.

Ce compte rendu peut aussi servir à prévoir des coupures pour diverses raisons telles que la maintenance ou l'essai. Les coupures sont proposées par le système DCS dans un intervalle de temps pendant lequel l'opération peut être lancée. Des copies de cet objet peuvent être échangées plusieurs fois entre le centre de conduite et la centrale électrique, jusqu'à ce qu'une heure de début réelle soit établie.

Une fois que le programme pour une opération de maintenance ou d'essais a été établi et accepté, aucun échange ultérieur n'est nécessaire, sauf si l'une des parties souhaite annuler ou renégocier l'opération. Dans ce cas, cette partie demande l'annulation du programme existant. Si l'opération doit être renégociée, un nouvel intervalle de temps peut être négocié pour son démarrage.

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 (identificateur de référence de disponibilité)

Identificateur unique à utiliser dans des références ultérieures au programme lors d'une révision.

TimeStamp (horodateur)

Heure à laquelle on envoie le compte rendu.

PlantReferenceID (identificateur de la centrale électrique)

Identificateur unique de la centrale électrique.

UnitID (identificateur de l'unité)

Identificateur unique de l'unité dont on rend compte.

ReportStatus (état du compte rendu)

Etat du compte rendu de disponibilité. PROPOSED (proposé) indique soit le premier programme, soit le programme révisé. Si l'état est PROPOSED, la date et l'heure de début et de fin renvoient aux dates de début et de fin les plus anciennes, avec une durée reflétant l'heure réelle estimée. CONFIRMED (confirmé) signale que le programme est accepté par le centre de conduite, auquel cas les dates de début et de fin renvoient aux dates planifiées. La durée peut ne pas être mentionnée. CANCELLED (annulé) s'explique par lui-même.

StartDateAndTime (date et heure de début)

Première date et heure de début (pour PROPOSED) ou date et heure planifiées (pour CONFIRMED).

StopDateAndTime (date et heure de fin)

Première date et heure de fin (pour PROPOSED) ou date et heure planifiées (pour CONFIRMED).

Duration (durée)

Intervalle de temps couvert par le compte rendu.

AvailabilityStatus (état de disponibilité)

AVAILABLE (disponible) indique que l'unité est prête à générer de l'énergie. UNAVAILABLE (indisponible) indique que l'unité est déconnectée et n'est pas prête à être planifiée.

EconomicImpact (impact économique)

Si la valeur est TRUE (vrai), indique qu'il y a un impact sur le tarif dans ce compte rendu.

PricImpact (impact sur le prix)

Prix associé à l'énergie générée par l'unité pendant la période couverte par le compte rendu.

RampRateImpact (impact sur le taux de croissance)

Si la valeur est TRUE, indique qu'il y a un impact sur le taux de croissance dans le compte rendu.

MaxRampRateUp (taux maximal de croissance)

Taux de croissance maximal prévu (MW) possible pour l'unité pendant l'intervalle de temps.

MaxRampRateDown (taux maximal de diminution)

Taux maximal de diminution prévu (MW) possible pour l'unité pendant l'intervalle de temps.

CapacityImpact (impact sur la capacité)

Si la valeur est TRUE, indique qu'il y a un impact sur la capacité associée à ce compte rendu.

UnitCapacity (capacité de l'unité)

Indique si l'unité concernée par le compte rendu est GROSS (brute) (sortie directe de l'unité), NET (nette) (sortie nette sur la ligne) ou BOTH (les deux).

GrossMaxCapacity (capacité maximale brute)

Capacité maximale brute opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

GrossMinCapacity (capacité minimale brute)

Capacité minimale brute opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

NetMaxCapacity (capacité maximale nette)

Capacité maximale nette opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

NetMinCapacity (capacité minimale nette)

Capacité minimale nette opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

TypeOfAvailability (type de disponibilité)

STANDBY (en attente) indique que l'unité est prête à générer de l'énergie mais qu'elle n'est pas en ligne. ONLINE (en ligne) indique que l'unité est synchronisée et qu'elle génère de l'énergie.

TimeToOnline(délai de mise en ligne)

Temps nécessaire pour que l'unité soit en ligne.

LFC

Si la valeur est TRUE, indique que l'unité peut recevoir des commandes et préciser le type de commandes. Les composants de LFC ci-dessous décrivent la disponibilité de l'unité pour différents types de commandes. Chaque type d'utilisation demande un tarif différent (c'est-à-dire qu'une unité de régulation sera facturée à un tarif différent de celui d'une unité de base). Ces composants peuvent être utilisés dans des combinaisons variées.

Dispatchable (distribuable)

Prêt à être exploité pour une valeur de consigne désignée extérieurement.

Regulating (régulateur)

Prêt à être exploité dans le but de réduire les ACE (Area Control Error = erreur de commande de zone) via le LFC d'un GCS ou d'un EMS.

Manually_Loaded (chargé manuellement)

Prêt pour des commandes locales.

ReasonForNoLFC (raison pour pas de LFC)

Raison pour laquelle l'unité ne peut pas être commandée en fréquence/puissance. STARTUP (démarrage) indique que l'unité n'est pas encore arrivée au point opérationnel désiré. UNSTABLE (instable) indique que l'unité n'est pas dans un état opérationnel acceptable pour le LFC.

ReasonForUnavailable (raison de l'indisponibilité)

Si l'unité est hors service, la raison peut être FORCED (forcée = coupure contrôlée mais non planifiée), SCHEDULED (planifiée = coupure à des fins de maintenance ou autre) ou TESTING (en essai).

ProvidingReserve (en réserve)

Si YES (oui), l'unité est actuellement en réserve.

Comment (commentaire)

Chaîne de texte contenant jusqu'à 256 caractères permettant d'ajouter une description définie par l'utilisateur au compte rendu.

A.3.3 Objet Real Time Status (état en temps réel)

L'objet suivant représente un compte rendu de la centrale électrique, soit au GCS, soit à un centre de conduite EMS déclarant le mode global d'opération de l'unité au moment du compte rendu. Le compte rendu en temps réel diffère du compte rendu de disponibilité car:

- il définit l'état réel d'une unité et pas une prévision;
- il inclut les attributs qui rendent compte des blocs externes, le cas échéant.

Ces comptes rendus en temps réel peuvent être émis à tout moment par le système DCS de la centrale électrique pour signaler les événements d'une unité.

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 (identificateur de référence de l'état temps réel)

Identificateur unique servant aux références ultérieures à ce rapport.

TimeStamp (horodateur)

Heure à laquelle on envoie le compte rendu.

PlantReferenceID (identificateur de référence de la centrale électrique)

Identificateur unique de la centrale électrique.

UnitID (identificateur de l'unité)

Identificateur unique de l'unité dont on rend compte.

AvailabilityStatus (état de disponibilité)

AVAILABLE (disponible) indique que l'unité est prête à générer de l'énergie. UNAVAILABLE (indisponible) indique que l'unité est déconnectée et n'est pas prête à être planifiée.

MaxRampRateUp (taux maximal de croissance)

Taux de croissance maximal prévu (MW) possible pour l'unité pendant l'intervalle de temps.

MaxRampRateDown (taux maximal de diminution)

Taux maximal de diminution prévu (MW) possible pour l'unité pendant l'intervalle de temps.

UnitCapacity (capacité de l'unité)

Indique si l'unité concernée par le compte rendu est GROSS (brute) (sortie directe de l'unité), NET (nette) (sortie nette sur la ligne) ou BOTH (les deux).

GrossMaxCapacity (capacité maximale brute)

Capacité maximale brute opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

GrossMinCapacity (capacité minimale brute)

Capacité minimale brute opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

NetMaxCapacity (capacité maximale nette)

Capacité maximale nette opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

NetMinCapacity (capacité minimale nette)

Capacité minimale nette opérationnelle prévue (MW) de sortie pendant l'intervalle de temps.

TypeOfAvailability (type de disponibilité)

STANDBY (en attente) indique que l'unité est prête à générer de l'énergie mais qu'elle n'est pas en ligne. ONLINE (en ligne) indique que l'unité est synchronisée et qu'elle génère de l'énergie.

TimeToOnline (délai de mise en ligne)

Temps nécessaire pour que l'unité soit en ligne.

LFC

Si la valeur est TRUE, indique que l'unité est commandée par la fréquence de la ligne et précise le type de commandes. Les composants de LFC ci-dessous décrivent les différents types de commandes. Ces composants peuvent être utilisés dans des combinaisons variées.

Dispatched (distribué)

Exploité par un point désigné extérieurement.

Regulating (régulateur)

Prêt à être exploité dans le but de réduire les ACE (Area Control Error = erreur de régulation de zone) via le LFC d'un GCS ou d'un EMS.

Manually_Loaded (chargé manuellement)

Prêt pour des commandes locales.

ReasonForNoLFC (raison pour pas de LFC)

Raison pour laquelle l'unité ne peut pas être commandée en fréquence/puissance. STARTUP (démarrage) indique que l'unité n'est pas encore arrivée au point opérationnel désiré. UNSTABLE (instable) indique que l'unité n'est pas dans un état opérationnel acceptable pour le LFC.

ExternallyBlockedHigh (bloqué extérieurement vers le haut)

Si la valeur est TRUE, l'unité est temporairement incapable d'augmenter sa sortie.

ExternallyBlockedLow (bloqué extérieurement vers le bas)

Si la valeur est TRUE, l'unité est temporairement incapable de diminuer sa sortie.

ReasonForUnavailable (raison de l'indisponibilité)

Si l'unité est hors service, la raison peut être FORCED (forcée = coupure contrôlée mais non planifiée), SCHEDULED (programmée, à des fins de maintenance ou autre), TESTING (en essai) ou EQUIPMENT (panne de l'équipement).

ProvidingReserve (en réserve)

Si YES (oui), l'unité est actuellement en réserve.

A.3.4 Objet Forecast Schedule (prévision de planification des révisions)**A.3.4.1 Généralités**

Le modèle d'objet ci-dessous représente une prévision du programme souhaité, soit d'un GCS, soit d'un centre de conduite EMS pour la centrale électrique. Le programme se compose d'un MW par rapport au temps, ainsi que du mode d'opération attendu (LFC_Component) en vigueur. Ces prévisions peuvent être soit à court terme (heures), soit à long terme (jours).

Ce modèle d'objet fournit de la souplesse en ce sens que l'information est contenue dans le programme. On utilise un objet matrice à usage général qui fournit des colonnes définies par l'utilisateur. On s'attend à ce qu'au moins deux colonnes soient utilisées pour fournir les valeurs MW ainsi que le mode d'opération LFC. Les modes LFC attendus sont Manual

(manuel), Dispatched (réparti), Regulating-Manual (régulation manuelle) et Regulating-Dispatched (régulation répartie), mais ce ne sont que des exemples. On peut utiliser un code pour représenter chaque mode de façon unique. Chaque ligne représente un incrément temporel.

Object: ForecastSchedule

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

ForecastScheduleReferenceID (identificateur de référence du programme de prévision)
 Identificateur unique utilisé pour les références ultérieures.

PlantReferenceID (identificateur de référence de la centrale électrique)
 Identificateur unique de la centrale électrique.

UnitID (identificateur de l'unité)

Identificateur unique de l'unité dont on rend compte.

ForecastType (type de prévision)

Décrit l'utilisation prévue de l'unité. GENERATION indique un programme pour la charge de base. RESERVE indique une utilisation pour la capacité de réserve uniquement.

StartTime (heure de début)

Heure de début du programme prévu.

PeriodResolution (découpage de la période)

Incrément de temps entre deux valeurs de la prévision.

NumberOfPeriods (nombre de périodes)

Nombre de valeurs de la prévision.

ListOfForecasts (liste des prévisions)

Valeurs MW planifiées pour chaque incrément temporel et mode d'opération LFC à chaque incrément temporel.

A.3.4.2 Objet Curve (courbe)

Le modèle d'objet ci-dessous représente un compte rendu par une centrale électrique, soit vers un GCS, soit vers un centre de conduite EMS d'une nouvelle courbe pour le calcul de choses telles que le taux de chaleur, la capacité MVAR ou le coût. La méthode de génération de la courbe est traitée localement. La courbe est représentée par une suite de segments de courbes, chacun d'eux étant défini sous la forme d'un polynôme. Chaque polynôme est à son tour représenté par une suite de coefficients pour chaque terme du polynôme.

Object: CurveSegmentDescription

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

Order (ordre)

Ordre du polynôme qui représente le segment de courbe.

LowRange (bas de l'intervalle)

Début de l'intervalle de la courbe représenté par le segment. A noter que si le LowRange est au-dessus du HighRange de la CurveSegmentDescription précédente, la courbe est indéfinie pour l'intervalle compris entre les deux.

HighRange (haut de l'intervalle)

Fin de l'intervalle de la courbe représenté par le segment. Il faut que le **HighRange** soit plus grand que le **LowRange** du segment.

Sequence of Coefficients (suite de coefficients)

Liste de nombres en virgule flottante de longueur égale à **Order** correspondant aux coefficients du polynôme qui représente le segment de la courbe. Si le segment de la courbe est représenté par $A_0 + A_1x + A_2x^2 \dots A_nx^n$, où n est l'ordre du polynôme, alors le premier élément de la liste représente A_0 et le dernier représente 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 (nom de la courbe)

Nom unique de la courbe.

UnitID (identificateur de l'unité)

Désignation de l'unité de génération renvoyant une modification de la courbe.

CurveType (type de la courbe)

Identifie la courbe normalisée des centrales électriques faisant l'objet du compte rendu. Il reste à déterminer le jeu complet de types de courbes. Le jeu initial défini est: HEAT_RATE, IO, IHR, MVAR_CAP, COST, OPACITY, SOX, NOX, CO2, USER-DEFINED.

NumberOfSegments (nombre de segments)

Nombre de segments (intervalles de la courbe) nécessaires à la représentation de la courbe.

Sequence of CurveSegmentDescription (suite de descriptions des segments de la courbe)

Liste des descriptions de segments de longueur **NumberOfSegments**, correspondant aux polynômes représentant chaque intervalle. Le premier segment correspond à l'intervalle le plus petit, le dernier à l'intervalle le plus grand.

A.3.4.3 Objets Power System Dynamics (dynamique des systèmes de puissance)

Les éléments de données supplémentaires ci-dessous sont nécessaires pour transmettre les communications de la dynamique du système de puissance entre la centrale électrique et le système GCS ou le système EMS du centre de conduite. Ces grandeurs scalaires peuvent être représentées par des objets Valeur de Donnée simples de TASE.2.

Information par unité de production

Coût

Transport/«pool» du centre de sécurité

Paramètres électriques

Volts

Watts

Vars

Réactance, Transitoire, Sub-transitoire

Niveau d'excitation

Paramètres mécaniques

Masse tournante

Caractéristiques du régulateur

A.4 Objet General Data Report

A.4.1 Généralités

L'objet GeneralDataReport est un objet conteneur qui peut être utilisé pour l'échange de données de compte rendu avec les zones de contrôle via le bloc 8 TASE2. Cet objet est destiné à l'émission d'une ou de plusieurs matrices de données. Chaque matrice peut contenir une ou plusieurs lignes de données avec un ou plusieurs en-têtes de colonne définis par l'utilisateur. Contrairement à l'objet Transfer Account défini dans la spécification de l'IEC 60870-6-802, il n'existe aucune relation temporelle entre les lignes.

Cet objet permet de transférer jusqu'à deux différentes matrices de valeur à virgule flottante, nombre entier ou texte. Les deux matrices de chaque type peuvent différer par le nombre de colonnes (matrices) de valeurs. Tous les types de données, y compris le type de données de matrice, sont des types de données normalisés définis dans l'IEC 60870-6-802, à l'exception d'un nouveau type de données de matrice texte et TransactionCode. Sinon, les seules différences par rapport à la norme sont le modèle abstrait et les définitions de structure.

A.4.2 Objet General Data Request

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 (numéro de référence du compte rendu de données général)

L'attribut GeneralDataReportReferenceNumber spécifie une valeur de référence unique entre l'émetteur et le destinataire pour identifier ce compte rendu particulier.

ReportName (nom du compte rendu)

Identificateur du texte de ce compte rendu.

ReportDateAndTime (date et heure du compte rendu)

L'attribut ReportDateAndTime permet au destinataire de connaître la date et l'heure de l'envoi du message par l'émetteur. Il ne concerne pas l'heure à laquelle s'appliquent les données du compte rendu.

TransactionCode (code de transaction)

L'attribut TransactionCode indique l'état de la transaction GeneralDataReport. Les valeurs sont données ci-dessous:

TransactionCode	Description
NEW	Soumission initiale d'une nouvelle proposition de programme ou de compte rendu
REQUESTED	Demande du compte rendu ou du programme via l'opération Account Request
REVISED	Version révisée du programme ou du compte rendu précédemment transmis
CONFIRMED	Le programme ou le compte rendu est à présent programmé pour mise en œuvre réelle sans nécessiter d'autres cycles d'approbation. Une opération de <i>curtailment (compression)</i> , <i>halt (arrêt)</i> , <i>revision (révision)</i> ou <i>deletion (annulation)</i> après cette étape nécessite de transmettre un nouveau programme ou compte rendu
DELETED	Tout moment auquel l'émetteur du programme ou du compte rendu annule la transaction (par exemple, du fait d'une panne de l'équipement)

NumberOfLocal Reference (nombre de références locales)

Cet attribut contient le nombre d'entrées de la ListOfLocalReference.

ListOfLocalReference (liste des références locales)

Cette liste contient les paramètres définis localement qui spécifient comment l'objet General Data Report est à traiter comme un ensemble. La signification de la liste des paramètres est traitée localement et fait l'objet d'un accord entre les parties intéressées pour chaque type d'objet General Data Report, pour tous les objets General Data Report ou pour une combinaison des deux.

NumberOfFloatingPoint1 (nombre de virgules flottantes 1)

L'attribut NumberOfFloatingPoint1 est le nombre de colonnes FloatingPoint1 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de FloatingPoint1 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfFloatingPoint1Rows (nombre de lignes à virgule flottante 1)

L'attribut NumberOfFloatingPoint1Rows est le nombre d'entrées (lignes) dans chaque colonne FloatingPoint1 dans ce compte rendu. Le nombre d'éléments dans la matrice FloatingPoint1 est (NumberOfFloatingPoint1Rows × NumberOfFloatingPoint1).

ListOfFloatingPoint1Values (liste des valeurs de virgule flottante 1)

L'attribut ListOfFloatingPoint1Values est la série de valeurs réelles de virgule flottante dans la matrice FloatingPoint1.

NumberOfFloatingPoint2 (nombre de virgules flottantes 2)

L'attribut NumberOfFloatingPoint2 est le nombre de colonnes FloatingPoint2 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de FloatingPoint2 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfFloatingPoint2Rows (nombre de lignes à virgule flottante 2)

L'attribut NumberOfFloatingPoint2Rows est le nombre d'entrées (lignes) dans chaque colonne FloatingPoint2 dans ce compte rendu. Le nombre d'éléments dans la matrice FloatingPoint2 est (NumberOfFloatingPoint2Rows × NumberOfFloatingPoint2).

ListOfFloatingPoint2Values (liste des valeurs de virgule flottante 2)

L'attribut ListOfFloatingPoint2Values est la série de valeurs réelles de virgule flottante dans la matrice FloatingPoint2.

NumberOfInteger1 (nombre d'entier 1)

L'attribut NumberOfInteger1 est le nombre de colonnes Integer1 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de Integer1 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfInteger1Rows (nombre de lignes à entier 1)

L'attribut NumberOfInteger1Rows est le nombre d'entrées (lignes) dans chaque colonne Integer1 dans ce compte rendu. Le nombre d'éléments dans la matrice Integer1 est (NumberOfInteger1Rows × NumberOfInteger1).

ListOfInteger1Values (liste des valeurs d'entier 1)

L'attribut ListOfInteger1Values est la série de valeurs réelles d'entier dans la matrice Integer1.

NumberOfInteger2 (nombre d'entiers 2)

L'attribut NumberOfInteger2 est le nombre de colonnes Integer2 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de Integer2 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfInteger2Rows (nombre de lignes à entier 2)

L'attribut NumberOfInteger2Rows est le nombre d'entrées (lignes) dans chaque colonne Integer2 dans ce compte rendu. Le nombre d'éléments dans la matrice Integer2 est (NumberOfInteger2Rows × NumberOfInteger2).

ListOfInteger2Values (liste des valeurs d'entier 2)

L'attribut ListOfInteger2Values est la série de valeurs réelles d'entier dans la matrice Integer2.

NumberOfText1 (nombre de textes 1)

L'attribut NumberOfText1 est le nombre de colonnes Text1 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de Text1 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfText1Rows (nombre de lignes à texte 1)

L'attribut NumberOfText1Rows est le nombre d'entrées (lignes) dans chaque colonne Text1 dans ce compte rendu. Le nombre d'éléments dans la matrice Text1 est (NumberOfText1Rows × NumberOfText1).

ListOfText1Values (liste des valeurs de texte 1)

L'attribut ListOfText1Values est la série de valeurs réelles de text dans la matrice Text1.

NumberOfText2 (nombre de textes 2)

L'attribut NumberOfText2 est le nombre de colonnes Text2 dans ce compte rendu. Ce nombre est également le drapeau qui indique si les données de Text2 sont non INCLUDED = 0 ou NOTINCLUDED=0 dans ce compte rendu.

NumberOfText2Rows (nombre de lignes de texte 2)

L'attribut NumberOfText2Rows est le nombre d'entrées (lignes) dans chaque colonne Text2 dans ce compte rendu. Le nombre d'éléments dans la matrice Text2 est (NumberOfText2Rows × NumberOfText2).

ListOfText2Values (liste des valeurs de texte 2)

L'attribut ListOfText2Values est la série de valeurs réelles de text dans la matrice Text2.

A.4.3 Objet General Data Response

L'objet GeneralDataResponse sert à transmettre une réponse à l'objet Block 8 de GeneralDataReport ou tout autre objet Block 8 nécessitant un accusé de réception. Il supporte également d'autres transactions impliquant plusieurs transferts de données.

Modèle d'objet Abstract

Object: **GeneralDataResponse**

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

ReportReferenceNumber (nombre de référence de compte rendu)

L'attribut ReportReferenceNumber spécifie une valeur de référence unique entre l'émetteur et le destinataire pour identifier cette réponse particulière. Ce nombre est le Reference Number du message reçu.

ReportName (nom du compte rendu)

Le ReportName (c'est-à-dire l'identificateur de texte) donné par le message reçu. Par exemple, pour une réponse à un message Generation Scheduling (planification de production), il s'agirait du nom de l'unité de production/programme donné par l'attribut Name du message reçu. Cet attribut associe également la réponse au message reçu.

NumberOfLocal Reference (nombre de références locales)

Cet attribut contient le nombre d'entrées de la ListOfLocalReference.

ListOfLocalReference (liste de références locales)

Cette liste contient les paramètres définis localement qui spécifient comment l'objet General Response est à traiter comme un ensemble. La signification de la liste des paramètres est traitée localement et fait l'objet d'un accord entre les parties intéressées pour chaque type

d'objet General Data Report, pour tous les objets Compte de Transfert ou pour une combinaison des deux.

ReportTimeStamp (horodateur du compte rendu)

Le ReportTimeStamp indique la date et l'heure contenues dans le message reçu. Cet attribut associe la réponse au message reçu.

ResponseData (données de réponse)

ResponseData constitue les informations supplémentaires fournies par l'application utilisées pour associer la réponse aux informations données par le message reçu.

ResponseCode (code de réponse)

Le ResponseCode est le code de réponse numérique associé à la réponse. Il a les valeurs qui correspondent aux réponses autorisées au TransactionCode définies dans l'objet GeneralDataReport. Les valeurs sont indiquées ci-dessous:

ResponseCode	Description
RECEIVED	Accusé de réception de la bonne réception du programme ou du compte rendu
APPROVED	Accusé de réception de l'approbation du compte rendu ou du programme
REJECTED	Accusé de réception du rejet du compte rendu ou du programme avant CONFIRMATION
CURTAILED	Annulation du programme CONFIRMED avant application par l'autorité de sécurité pour assurer la fiabilité du réseau électrique ou pour d'autres raisons d'ordre opérationnel
HALT	Annulation du programme CONFIRMED déjà appliqué et en cours d'exécution par l'autorité de sécurité pour assurer la fiabilité du réseau électrique ou pour d'autres raisons d'ordre opérationnel

ResponseText (texte de réponse)

Le ResponseText est le texte lisible fourni par l'application associé à la réponse.

Annexe B (informative)

Types d'objets MMS TASE.2 (2002) supplémentaires

B.1 Généralités

L'IEC 60870-6-802:2002 comprenait plusieurs modèles d'objets normatifs qui ne relèvent pas du domaine d'application de la présente version. L'intention est de déconseiller l'utilisation de ces éléments de modèles dans la prochaine révision de l'IEC 60870-6-802. Cependant, à titre historique, les informations de la version 2002 sont reproduites dans les Articles B.2 à B.5.

B.2 Types de Comptes de Transfert

Les types de base suivants sont référencés dans les descriptions de types complexes:

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 }

Descriptions du type TransferAccount

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

}

Descriptions du type **TransmissionSegment**

TATransmissionSegment STRUCTURE

{

COMPONENT	TransmissionReference	ReferenceNum,
COMPONENT	UtilWheeling	UtilityId,
COMPONENT	UtilPaying	UtilityId,
COMPONENT	TransmissionSegType	TransmissionSegCode,

COMPONENT UtilIn	UtilityId,
COMPONENT UtilOut	UtilityId,
COMPONENT InterchangePtIn	Interchangeld,
COMPONENT InterchangePtOut	Interchangeld,
COMPONENT InterchangePt	Interchangeld,
COMPONENT NumberFloatIds	NumFloats,
COMPONENT NumberIntegerIds	NumIntegers
}	

Descriptions du type Transmission Segment

TATransmissionSegmentProfile STRUCTURE

{	
COMPONENT TransmissionReference	ReferenceNum,
COMPONENT UtilWheeling	UtilityId,
COMPONENT UtilPaying	UtilityId,
COMPONENT TransmissionSegType	TransmissionSegCode,
COMPONENT UtilIn	UtilityId,
COMPONENT UtilOut	UtilityId,
COMPONENT InterchangePtIn	Interchangeld,
COMPONENT InterchangePtOut	Interchangeld,
COMPONENT InterchangePt	Interchangeld,
COMPONENT NumberProfileValues	NumProfiles
}	

Descriptions du type ProfileValue

TAProfileValue STRUCTURE

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

Descriptions du type Account Request

AccountRequest STRUCTURE

{	
COMPONENT ReferenceTar	ReferenceNum,
COMPONENT StartTime	ScheduleTime,
COMPONENT Duration	Duration,
COMPONENT RequestId	ReferenceNum,
COMPONENT TaConditionsRequested	TAConditions
}	

B.3 Descriptions du type Power Plant

Les types de base suivants sont utilisés dans les définitions de type «Power Plant – EMS Data Exchange»:

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
}		

Descriptions du type Availability Report**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	bitstring { price[0], ramp[1],
	capacity[2] },	
}		

```

    COMPONENT PriceImpact           Price,
    COMPONENT RampImpact            RampRates,
    COMPONENT CapacityImpact       UnitCapacity,
    COMPONENT Availability         AvailabilityClass,
    COMPONENT Comment              VisibleString { width 255 }
}

```

UnAvailable STRUCTURE

```

{
    COMPONENT AvailabilityReferenceld ReferenceNum,
    COMPONENT PlantReferenceld      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 }
}

```

Descriptions du type Real Time Status

StatusAvailable STRUCTURE

```

{
    COMPONENT AvailabilityReferenceld ReferenceNum,
    COMPONENT PlantReferenceld      PlantId,
    COMPONENT UnitId                 UnitId,
    COMPONENT TimeStamp              TimeStampS,
    COMPONENT RampStatus             RampRates,
    COMPONENT CapacityStatus        UnitCapacity,
    COMPONENT AvailabilityStatus    AvailabilityClass,
    COMPONENT ProvidingReserve       boolean
}

```

StatusUnAvailable STRUCTURE

```

{
    COMPONENT AvailabilityReferenceld ReferenceNum,
    COMPONENT PlantReferenceld      PlantId,
    COMPONENT UnitId                 UnitId,
    COMPONENT TimeStamp              TimeStampS,
    COMPONENT ReasonForUnavailable  AvailReasonCode,
    COMPONENT ProvidingReserve       boolean
}

```

Descriptions du type Forecast

Forecast STRUCTURE

```

{
    COMPONENT ForecastScheduleReferenceld ReferenceNum,
    COMPONENT PlantReferenceld          PlantId,
    COMPONENT UnitId                     UnitId,
    COMPONENT StartTime                  GMTBasedS,
    COMPONENT PeriodResolution          TimeIntervals,
}

```



```

        COMPONENT NumPeriods                integer,
        COMPONENT ForecastType             bitstring { generation[0], reserve[1] }
    }

```

Descriptions du type Curve

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 PlantReferenceld    PlantId,
    COMPONENT UnitId              UnitId,
    COMPONENT CurveType           CurveType,
    COMPONENT NumberOfSegments    NumSegs
}

```

B.4 Power System Dynamics (dynamique du système de puissance)

B.4.1 Généralités

Aucun nouveau type de données n'est nécessaire pour support Power System Dynamics. Ils correspondent aux variables des types de base MMS simples.

B.4.2 Type de données Matrix

Les types de données Matrix servent à représenter des suites ou des listes d'enregistrements logiques, dans lesquels chaque enregistrement logique contient un jeu de valeurs pour une

étape de la séquence. Un programme de transmission dans lequel on prévoit les grandeurs (énergie, capacité, etc.) ainsi que leur coût associé transmis à chaque intervalle de temps constitue un exemple de suite. Le type **Matrix** permet la représentation d'ensembles arbitraires de valeurs et des suites de longueur arbitraire, même quand on ne peut pas connaître avant l'exécution l'ensemble particulier de types de données, la sémantique ainsi que la longueur des suites.

La représentation de base d'une suite d'enregistrements logiques homogènes dans un compte rendu d'information MMS utilisant les Types de Données **Matrix** utilise la suite de Variables Nommées MMS suivantes:

- a) Il faut que le compte rendu contienne une variable *d'en-tête* qui est en général spécifique au modèle d'objet qui contient la suite ou la liste. La variable d'en-tête doit contenir la longueur attendue de cette suite ou de cette liste, ainsi qu'un comptage du nombre total de valeurs entières et le nombre total de valeurs en virgule flottante qui sont à transmettre pour chaque enregistrement de la suite. Le nombre et l'identité des valeurs entières et en virgule flottante peuvent changer d'un compte rendu à l'autre mais seront cohérents pour chaque enregistrement logique dans un compte rendu donné.

Si des valeurs en virgule flottante sont à inclure dans les enregistrements logiques:

Le compte rendu contiendra une variable nommée **Matrix_Id** de type **MatrixID** pour chaque valeur en virgule flottante à inclure dans les enregistrements logiques. Le nombre de variables **Matrix_Id** pour les valeurs en virgule flottante attendues peut être déterminé à partir de la variable d'en-tête.

Pour chaque enregistrement logique transmis, on inclut une variable de type **FloatArrayXX** contenant les valeurs en virgule flottante de l'enregistrement. La taille de **FloatArrayXX** (XX est la dimension du tableau) est arbitraire mais doit être supérieure ou égale au nombre de valeurs en virgule flottante à inclure. Si le tableau est plus grand que le nombre de valeurs en virgule flottante incluses dans les enregistrements logiques, le tableau est rempli à partir du rang inférieur (en partant de **FloatArray[0]**), le reste étant ignoré.

Si des valeurs entières sont à inclure dans les enregistrements logiques:

Le compte rendu contiendra une variable nommée **Matrix_Id** de type **MatrixID** pour chaque valeur entière à intégrer dans les enregistrements logiques. Le nombre de variables **Matrix_ID** pour les valeurs entières attendues peut être déterminé à partir de la variable d'en-tête.

Pour chaque enregistrement logique transmis, on inclut une variable de type **IntegerArrayXX** contenant les valeurs entières de l'enregistrement. La taille de **IntegerArrayXX** (XX est la dimension du tableau) est arbitraire mais doit être supérieure ou égale au nombre de valeurs entières à inclure. Si le tableau est plus grand que le nombre de valeurs entières incluses dans les enregistrements logiques, le tableau est rempli à partir du rang inférieur (en partant de **IntegerArray[0]**), le reste étant ignoré.

Si des valeurs en texte sont à inclure dans les enregistrements logiques:

Le compte rendu contiendra une variable nommée **Matrix_Id** de type **MatrixID** pour chaque valeur en texte à inclure dans les enregistrements logiques. Le nombre de variables **Matrix_ID** pour les valeurs en texte attendues peut être déterminé à partir de la variable d'en-tête.

Pour chaque enregistrement logique transmis, on inclut une variable de type **Text32ArrayXX** contenant les valeurs en texte de l'enregistrement. La taille de **Text32ArrayXX** (XX est la dimension du tableau) est arbitraire mais doit être supérieure ou égale au nombre de valeurs en texte à inclure. Si le tableau est plus grand que le nombre de valeurs en texte incluses dans les enregistrements logiques, le tableau est rempli à partir du rang inférieur (en partant de **Text32ArrayXX [0]**), le reste étant ignoré.

Chaque tuple de **FloatArrayXX**, **IntegerArrayXX** et **Text32ArrayXX** constitue un enregistrement logique (et par là une ligne de la matrice).

Les types suivants sont définis:

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}

où XX est une dimension arbitraire du tableau.

Exemple: tableau de 256 octets = IntegerArray256, tableau de 1024 octets = IntegerArray1024

B.5 Descriptions du type GeneralDataReport

La plupart des types de base référencés dans les descriptions des types complexes GeneralDataReport ci-dessous sont les mêmes que ceux utilisés dans les Transfer Accounts et sont définis dans la section relative aux types Transfer Account Types. Les nouveaux types de base suivants sont également référencés dans la description de type complexe GeneralDataReport:

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

Le type complexe suivant est utilisé pour l'échange des General Data Reports:

GeneralDataReport STRUCTURE

{		
COMPONENT	GeneralDataReportReferenceNumber	ReferenceNum
COMPONENT	ReportName	NameString
COMPONENT	ReportDateAndTime	TimeStamps
COMPONENTS	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
}		

B.6 Descriptions du type GeneralDataResponse

Le type complexe suivant est utilisé pour l'échange des 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

}

Annexe C (informative)

Correspondance des objets TASE.2 (2002) avec les types MMS

C.1 Généralités

L'IEC 60870-6-802:2002 comprenait plusieurs types MMS normatifs qui ne relèvent pas du domaine d'application de la présente version. L'intention est de déconseiller l'utilisation de ces types dans la prochaine révision de l'IEC 60870-6-802. Cependant, à titre historique, les informations de la version 2002 sont reproduites dans les articles suivants.

C.2 Correspondance avec les comptes de transfert

C.2.1 Correspondance avec TransferAccount

Les TransferAccount ne sont visibles qu'au travers des mécanismes Transfer Account Reporting (compte rendu de compte de transfert) et Account Query (requête de compte), comme définis dans l'IEC 60870-6-503. Les Transfer Accounts correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chacune des Variables Nommées MMS utilisées pour la correspondance du modèle d'objet Transfer Account n'est utilisée que dans les comptes rendus d'Information MMS et renvoie le Résultat d'Accès MMS OBJECT-ACCESS-DENIED (accès refusé) en lecture et en écriture.

Chacune des classes de Comptes de Transfert contient des informations d'en-tête et éventuellement une ou plusieurs listes de données (ListOfLocalReference, ListOfTransmissionSegment, ListOfProfileValues et ListOfPeriodicValues). Les listes de données sont représentées comme suit:

ListOfLocalReference – suite de Variables Nommées MMS, chacune nommée **Reference_Num** et de type **ReferenceNum**. Il doit y avoir une Variable Nommée MMS pour chaque entrée de la ListOfLocalReference.

ListOfTransmissionSegment – suite de Variables Nommées MMS. Pour chaque entrée de la ListOfTransmissionSegment:

Si le compte rendu est tenu de contenir des données périodiques:

pour chaque segment de transmission objet du compte rendu:

une Variable Nommée MMS **TA_Transmission_Segment**, de type **TATransmissionSegment**;

si des valeurs en virgule flottante sont transmises pour ce segment:

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en virgule flottante inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données.

Pour chaque période de temps transmise, une Variable Nommée MMS **Float_Array_XX**, de type **FloatArrayXX**;

s'il y a des valeurs entières transmises pour ce segment:

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en valeur entière inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données.

Pour chaque période de temps transmise, une Variable Nommée MMS **Integer_Array_XX**, de type **IntegerArrayXX**;

Si le compte rendu contient des données de profil:

Pour chaque Segment de transmission objet du compte rendu:

Une Variable Nommée MMS **TA_Transmission_Segment_Profile**, de type **TATransmissionSegmentProfile**

Pour chaque Valeur de Profil transmise, une Variable Nommée MMS **TA_Profile_Value**, de type **TAProfileValue**;

ListOfProfileValues – suite de Variables Nommées MMS, chacune appelée **TA_Profile_Value** et de type **TAProfileValue**, pour chaque élément de la ListOfProfileValues.

ListOfPeriodicValues – suite des Variables Nommées MMS ci-dessous. A noter que chaque variable FloatArray et chaque variable IntegerArray ci-dessous représentent un élément distinct de la ListOfPeriodicValues:

Si, pour chaque période, on fournit des valeurs en virgule flottante:

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en virgule flottante inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données.

Pour chaque période de temps transmise, une Variable Nommée MMS **Float_Array_XX**, de type **FloatArrayXX**;

Si, pour chaque période, on fournit des valeurs entières:

Une Variable Nommée MMS **Matrix_Id**, de type **MatrixId** pour chaque grandeur en valeur entière inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données.

Pour chaque période de temps transmise, une Variable Nommée MMS **Integer_Array_XX**, de type **IntegerArrayXX**

Voir B.4.2 pour une description plus détaillée de l'utilisation des types Matrix.

L'ordre de transmission des Variables Nommées MMS représentant le TransferAccount doit être le suivant:

- a) variable TAConditions_Detected (voir 7.1.4.4.4 de l'IEC 60870-6-503:2002);
- b) variable Request_Id, si le compte rendu est une réponse à une Opération de Requête (voir 7.1.5.1.2 de l'IEC 60870-6-503:2002);
- c) la variable d'en-tête Compte de Transfert (voir détails plus loin);
- d) variables représentant la ListOfLocalReference, s'il y en a;
- e) variables représentant la ListOfTransmissionSegments, s'il y en a;
- f) variables représentant la ListOfProfileValues ou la ListOfPeriodicValues.

La Variable Nommée MMS (et son type MMS) représentant les informations d'en-tête est choisie à partir de trois attributs de l'objet Transfer Account, comme défini dans le tableau ci-dessous:

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

Les autres combinaisons d'attributs ne sont pas supportées dans le modèle d'objet Transfer Account.

Les Variables nommées MMS utilisées ont les types suivants:

TA_Segments_Periodic	type TASEgmentsPeriodic
TA_NoSegments_Periodic	type TANOsegmentsPeriodic
TA_Segments_Profile	type TASEgmentsProfile
TA_NoSegments_Profile	type TANOsegmentsProfile

La correspondance spécifique des attributs de l'objet Transfer Account est la suivante:

TransferAccountReference

Correspond au COMPOSANT **TransferAccountRef** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**.

SendUtility

Correspond au COMPOSANT **SendUtility** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**. Le codage des valeurs spécifiques est déterminé par un accord entre le client et le serveur.

ReceiveUtility

Correspond au composant **RecvUtility** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**. Le codage des valeurs spécifiques est déterminé par un accord entre le client et le serveur.

SellingAgent

Correspond au composant **SellingUtility** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**. Le codage des valeurs spécifiques est déterminé par un accord entre le client et le serveur.

BuyingAgent

Correspond au COMPOSANT **BuyingUtility** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**. Le codage des valeurs spécifiques est déterminé par un accord entre le client et le serveur.

TimeStamp

Correspond au COMPOSANT **TimeStamp** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile** ou **TANOsegmentsProfile**.

TransactionCode

Correspond au COMPOSANT **TransactionCode** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile**, ou **TANOsegmentsProfile**.

NumberOfLocalReference

Correspond au COMPOSANT **NumberOfLocalReferences** d'une Variable MMS de type **TASEgmentsPeriodic**, **TANOsegmentsPeriodic**, **TASEgmentsProfile**, ou **TANOsegmentsProfile**.

ListOfLocalReference

Correspond à une suite de Variables Nommées MMS appelées chacune **Reference_Num** et de type **ReferenceNum**. Il doit y avoir une référence de Variable Nommée MMS dans le compte rendu pour chaque entrée de la ListOfLocalReference. Le nombre de références Reference_Num du compte rendu doit être cohérent avec la valeur de l'attribut NumberOfLocalReference.

Name

Correspond au COMPOSANT **Name** d'une variable MMS de type **TASegmentsPeriodic**, **TANoSegmentsPeriodic**, **TASegmentsProfile** ou **TANoSegmentsProfile**. Si l'attribut Name n'est pas présent, le COMPOSANT **Name** doit être une chaîne vide (NULL).

TransmissionSegmentOption

Pas de correspondance, mais utilisé lors du choix des types de données des Variables MMS (voir ci-dessus).

NumberOfTransSegments

Correspond au COMPOSANT **NumberSegments** d'une variable MMS de type **TASegmentsPeriodic** ou **TASegmentsProfile**.

ListOfTransmissionSegments

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de la ListOfTransmissionSegment:

- a) Incrire les paramètres définissant le segment. Si le compte rendu est tenu de contenir des données périodiques, alors, pour chaque segment de transmission objet du compte rendu, inscrire les paramètres définissant le segment en tant que valeurs d'une Variable Nommée MMS appelée **TA_Transmission_Segment** et de type **TATransmissionSegment** (Voir 7.2.2.1 pour plus de détails sur cette correspondance). Si le compte rendu est tenu de contenir des données de profil, alors, pour chaque segment de transmission objet du compte rendu, inscrire les paramètres définissant le segment de fichier en tant que valeurs d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** et de type **TATransmissionSegmentProfile** (Voir C.2.2.2 pour plus de détails sur cette correspondance).

- b) Incrire les données du segment:

Si le compte rendu est tenu de contenir des données périodiques, alors, pour chaque période:

Si, pour ce segment, on fournit des valeurs en virgule flottante:

Incrire les identificateurs des valeurs en virgule flottante en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Incrire les valeurs en virgule flottante pour la période de temps concernant ce segment dans une Variable Nommée MMS appelée **Float_Array_XX** et de type **FloatArrayXX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

Si, pour ce segment, on fournit des valeurs entières:

Incrire les identificateurs des valeurs entières en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Inscrire les valeurs entières pour la période de temps ou concernant ce segment dans une Variable Nommée MMS appelée **Integer_Array_XX** et de type **IntegerArrayXX**, où XX est assez grand pour contenir toutes les valeurs entières.

Si le compte rendu est tenu de contenir des données de profil, alors pour chaque période de temps faisant l'objet du compte rendu:

Inscrire les données de profil concernant ce segment dans une Variable Nommée MMS appelée **TA_Profile_Value** et de type **TAProfileValue**.

DataType

Pas de correspondance, mais utilisé lors du choix des types de données des Variables MMS (voir ci-dessus).

StartTime

Correspond au COMPOSANT **StartTime** d'une Variable MMS de type **TASegmentsPeriodic** ou **TANoSegmentsPeriodic**.

PeriodResolution

Correspond au COMPOSANT **PeriodResolution** d'une Variable MMS de type **TASegmentsPeriodic** ou **TANoSegmentsPeriodic**.

NumberOfPeriods

Correspond au COMPOSANT **NumberPeriods** d'une Variable MMS de type **TASegmentsPeriodic** ou **TANoSegmentsPeriodic**.

ListOfPeriodicValues

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de la ListOfPeriodicValues:

a) si on fournit des valeurs en virgule flottante:

Inscrire les identificateurs des valeurs en virgule flottante en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque période, inscrire les valeurs comme les valeurs d'une Variable Nommée MMS appelée **Float_Array_XX** et de type **FloatArrayXX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

b) si on fournit des valeurs entières:

Inscrire les identificateurs des valeurs entières en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque période, inscrire les valeurs comme les valeurs d'une Variable Nommée MMS appelée **Integer_Array_XX** et de type **IntegerArrayXX**, où XX est assez grand pour contenir toutes les valeurs entières.

NumberOfProfiles

Correspond au COMPOSANT **NumberOfProfiles** d'une Variable MMS de type **TASegmentsProfile** ou **TANoSegmentsProfile**.

ListOfProfileValues

Correspond, pour chaque élément de la ListOfProfileValues, à une suite de Variables Nommées MMS appelées chacune **TA_Profile_Value** et de type **TAProfileValue**. Le nombre de références de **TA_Profile_Value** du compte rendu doit être cohérent avec la valeur de l'attribut **NumberOfProfiles**.

C.2.2 Correspondance de TransmissionSegment (segment de transmission)

C.2.2.1 TATransmissionSegment

Chaque objet TansmissionSegment correspond à une suite de Variables Nommées MMS. La première variable s'appelle **TA_Transmission_Segment** et est de type **TATransmissionSegment**. Elle fournit les paramètres (identificateurs de référence, points d'échange, etc.) du segment. Le reste des variables dépend des types de données des valeurs transmises pour chaque segment. Les variables sont décrites dans l'article précédent.

Les attributs de définition du segment correspondent comme suit:

TransmissionReference

Correspond au COMPOSANT **TransmissionReference** d'une variable MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

UtilWheeling

Correspond au COMPOSANT **UtilWheeling** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

UtilPaying

S'il est présent, correspond au COMPOSANT **UtilityPaying** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

TransmissionSegType

Correspond au COMPOSANT **TransmissionSegType** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**. La valeur est codée comme suit: 0=DIRECT, 1=INONLY, 2=OUTONLY, 3=INOUT.

UtilIn

S'il est présent, correspond au COMPOSANT **UtilIn** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

InterchangePtIn (point d'échange en entrée)

S'il est présent, correspond au COMPOSANT **InterchangePtIn** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

UtilOut

S'il est présent, correspond au COMPOSANT **UtilOut** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

InterchangePtOut

S'il est présent, correspond au COMPOSANT **InterchangePtOut** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

InterchangePt

S'il est présent, correspond au COMPOSANT **InterchangePtIn** d'une Variable Nommée MMS appelée **TA_Transmission_Segment** de type **TATransmissionSegment**.

NumberFloatIds

Le nombre de Float Ids dans **ListOfSegmentData** pour ce **TransmissionSegment**.

NumberIntegerIds

Le nombre de Integer Ids dans **ListOfSegmentData** pour ce **TransmissionSegment**.

ListOfSegmentData

Correspond aux Variables Nommées MMS de type **MatrixId**, **FloatArrayXX** et **IntegerArrayXX**, comme défini ci-dessus.

C.2.2.2 TATransmissionSegmentProfile

Chaque objet **TransmissionSegment** correspond à une suite de Variables Nommées MMS. La première variable est appelée **TA_Transmission_Segment_Profile** et est de type **TATransmissionSegmentProfile**. Elle fournit les paramètres (identificateurs de référence, points d'échange, etc.) du segment. Le reste des variables dépend des types de données des valeurs transmises pour chaque segment. Les variables sont décrites dans l'article précédent.

Les attributs de définition du segment correspondent comme suit:

TransmissionReference

Correspond au COMPOSANT **TransmissionReference** d'une variable MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

UtilWheeling

Correspond au COMPOSANT **UtilWheeling** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

UtilPaying

S'il est présent, correspond au COMPOSANT **UtilityPaying** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

TransmissionSegType

Correspond au COMPOSANT **TransmissionSegType** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**. La valeur est codée comme suit: 0=DIRECT, 1=INONLY, 2=OUTONLY, 3=INOUT.

UtilIn

S'il est présent, correspond au COMPOSANT **UtilIn** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

InterchangePtIn

S'il est présent, correspond au COMPOSANT **InterchangePtIn** d'une variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

UtilOut

S'il est présent, correspond au COMPOSANT **UtilOut** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

InterchangePtOut

S'il est présent, correspond au COMPOSANT **InterchangePtOut** d'une Variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

InterchangePt

S'il est présent, correspond au COMPOSANT **InterchangePtIn** d'une variable Nommée MMS appelée **TA_Transmission_Segment_Profile** de type **TATransmissionSegmentProfile**.

NumberProfileValues

Le nombre d'objets **ProfileValue** dans **ListofSegmentData** pour ce **TransmissionSegment**

ListOfSegmentData

Correspond aux Variables Nommées MMS de type **ProfileValue**, comme défini ci-dessus.

C.2.3 Correspondance de ProfileValue

RampStartTime

Correspond au COMPOSANT **RampStartTime** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

RampDuration

Correspond au COMPOSANT **RampDuration** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

ProfilePrice

Correspond au COMPOSANT **ProfilePrice** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

TargetClass

Correspond au COMPOSANT **ProfileTargetClass** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**. La valeur est codée comme suit: 0=OTHER, 1=ENERGY, 2=CAPACITY.

ProfileEnergy

S'il est présent, correspond au COMPOSANT **ProfileTarget** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

ProfileCapacity

S'il est présent, correspond au COMPOSANT **ProfileTarget** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

ProfileOther

S'il est présent, correspond au COMPOSANT **ProfileTarget** d'une Variable Nommée MMS appelée **TA_Profile_Value** de type **TAProfileValue**.

C.2.4 Correspondance de AccountRequest

Le modèle d'objet AccountRequest correspond à une variable MMS de type AccountRequest. La correspondance des attributs est:

AccountRequestName

Correspond à l'identificateur d'une variable MMS écrite pour générer la demande.

TransferAccountReference

Correspond au COMPOSANT **ReferenceTar** d'une variable MMS de type **AccountRequest**.

StartTime

Correspond au COMPOSANT **StartTime** d'une variable MMS de type **AccountRequest**.

Duration

Correspond au COMPOSANT **Duration** d'une variable MMS de type **AccountRequest**.

RequestId

Correspond au COMPOSANT **RequestId** d'une variable MMS de type **AccountRequest**.

TaConditionsRequested

Correspond au COMPOSANT **TaConditionsRequested** d'une Variable MMS de type **AccountRequest**.

C.3 Correspondance de Power Plant

C.3.1 Disponibilité de Report Mapping

La Variable Nommée MMS (et son type MMS) représentant le Availability Report est sélectionnée à partir de l'attribut Availability Status de l'objet Availability Report. Si l'attribut Availability Status est AVAILABLE, le compte rendu correspond à la Variable Nommée MMS appelée **Available** et de type **Available**. Si l'attribut Availability Status est UNAVAILABLE, le compte rendu correspond à la Variable Nommée MMS appelée **UnAvailable** et de type **UnAvailable**.

Les objets Availability Report correspondent à une suite d'une ou plusieurs Variables Nommées MMS mais seulement de façon instantanée pendant la génération du compte rendu.

Les Variables Nommées MMS servant à la correspondance du modèle d'objet AvailabilityReport ne sont utilisées que dans les MMS Information Reports et renvoient un MMS Access Result OBJECT-ACCESS-DENIED en lecture et en écriture.

Les autres attributs d'objet correspondent comme suit:

AvailabilityReferenceID (identificateur de référence de disponibilité)

Correspond au COMPOSANT **AvailabilityReferenceId** d'une Variable Nommée MMS de nom et de type sélectionnés.

Timestamp

Correspond au COMPOSANT **Timestamp** d'une Variable Nommée MMS de nom et de type sélectionnés.

PlantReferenceID (identificateur de référence de la centrale électrique)

Correspond au COMPOSANT **PlantReferenceId** d'une Variable Nommée MMS de nom et de type sélectionnés.

UnitID

Correspond au COMPOSANT **UnitID** d'une Variable Nommée MMS de nom et de type sélectionnés.

ReportStatus

Correspond au COMPOSANT **ReportStatus** d'une Variable Nommée MMS de nom et de type sélectionnés, dont les significations sont les suivantes: 0=PROPOSED, 1=CONFIRMED, 2=CANCELLED.

StartDateAndTime

Correspond au composant **StartDateAndTime** d'une Variable Nommée MMS de nom et de type sélectionnés.

StopDateAndTime

Correspond au composant **StopDateAndTime** d'une Variable Nommée MMS de nom et de type sélectionnés.

Duration

Correspond au COMPOSANT **Duration** d'une Variable Nommée MMS de nom et de type sélectionnés.

Availability Status

L'attribut Availability Status sert à la sélection du nom et du type de la Variable Nommée MMS représentant le compte rendu.

EconomicImpact

Correspond au COMPOSANT **Impact** de la Variable Nommée MMS de type **Available** avec les significations suivantes: 1=YES, 0=NO.

PricelImpact

S'il est présent, correspond au COMPOSANT **PricelImpact** de la Variable Nommée MMS de type **Available**. Si le bit 0 du COMPOSANT **Impact** est NO, le COMPOSANT **PricelImpact** doit être ignoré.

RampRateImpact

Correspond au bit 1 du COMPOSANT **Impact** d'une Variable Nommée MMS de type **Available** avec les significations suivantes: 1=YES, 0=NO.

MaxRampRateUp

S'il est présent, correspond au champ **MaxRampRateUp** du COMPOSANT **RampImpact** de la Variable Nommée MMS de type **Available**. Si le bit 1 du COMPOSANT **Impact** est NO, le COMPOSANT **MaxRampRateUp** doit être ignoré.

MaxRampRateDown

S'il est présent, correspond au champ **MaxRampRateDown** du COMPOSANT **RampImpact** de la Variable Nommée MMS de type **Available**. Si le bit 1 du COMPOSANT **Impact** est NO, le COMPOSANT **MaxRampRateDown** doit être ignoré.

CapacityImpact

Correspond au bit 2 du COMPOSANT **Impact** d'une Variable Nommée MMS de type **Available** avec les significations suivantes: 1=YES, 0=NO.

Chacun des attributs suivants (quand il est présent) correspond à un champ du COMPOSANT **CapacityImpact** (type **UnitCapacity**) de la Variable Nommée MMS de type **Available**. Si le bit 2 du COMPOSANT **Impact** est NO, tous les champs du COMPOSANT **CapacityImpact** doivent être ignorés.

UnitCapacity

S'il est présent, correspond au COMPOSANT **CapacityClass** COMPONENT de type **UnitCapacity** avec les significations suivantes: bit 0 = 1 implique GROSS, bit 1 = 1 implique NET, bit 0 et bit 1 = 1 implique BOTH.

GrossMaxCapacity

S'il est présent, correspond au COMPOSANT **GrossMaxCapacity** de type **UnitCapacity**. Si le bit 0 du COMPOSANT **CapacityClass** n'est pas = 1, **GrossMaxCapacity** doit être ignoré.

GrossMinCapacity

S'il est présent, correspond au COMPOSANT **GrossMinCapacity** de type **UnitCapacity**. Si le bit 0 du COMPOSANT **CapacityClass** n'est pas = 1, **GrossMinCapacity** doit être ignoré.

NetMaxCapacity

S'il est présent, correspond au COMPOSANT **NetMaxCapacity** de type **UnitCapacity**. Si le bit 1 du COMPOSANT **CapacityClass** n'est pas = 1, **NetMaxCapacity** doit être ignoré.

NetMinCapacity

S'il est présent, correspond au COMPOSANT **NetMinCapacity** de type **UnitCapacity**. Si le bit 1 du COMPOSANT **CapacityClass** n'est pas = 1, **NetMinCapacity** doit être ignoré.

Chacun des attributs suivants (quand il est présent) correspond à un champ du COMPOSANT **Availability** (type **AvailabilityClass**) de la Variable Nommée MMS de type **Available**:

TypeOfAvailability

Correspond au bit 0 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=STANDBY, 1=ONLINE.

TimeToOnline

Correspond au COMPOSANT **TimeToOnline** de type **AvailabilityClass**. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 1, cet attribut doit être ignoré.

LFC

Correspond au bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES.

Dispatchable

Correspond au bit 2 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

Regulating

Correspond au bit 3 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

Manually_Loaded

Correspond au bit 4 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

ReasonForNoLFC

Correspond au COMPOSANT **ReasonForNoLFC** de type **AvailabilityClass** avec les significations suivantes: 0=STARTUP, 1=UNSTABLE. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 1, cet attribut doit être ignoré.

ReasonForUnavailable

Correspond au COMPOSANT **ReasonForUnavailable** de la Variable Nommée MMS de type **UnAvailable** avec les significations suivantes: 0=FORCED, 1=SCHEDULED, 2=TESTING.

ProvidingReserve

Correspond au COMPOSANT **ProvidingReserve** de la Variable Nommée MMS sélectionnée (type **Available** ou **UnAvailable**) avec les significations suivantes: 0=NO, 1=YES.

Comment

Correspond au COMPOSANT **Comment** de la Variable Nommée MMS sélectionnée (type **Available** ou **UnAvailable**).

C.3.2 Correspondance de Real Time Status

La Variable Nommée MMS (et son type MMS) représentant le compte rendu «Real Time Status» est sélectionnée en fonction de l'attribut Availability Status de l'objet «Real Time Status». Si l'attribut Availability Status est AVAILABLE, le compte rendu correspond à une Variable Nommée MMS appelée **Status_Available** et de type **StatusAvailable**. Si l'attribut Availability Status est UNAVAILABLE, le compte rendu correspond à une Variable Nommée MMS appelée **Status_UnAvailable** de type **StatusUnAvailable**.

Les objets Real Time Status correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chaque Variable Nommée MMS utilisée pour la correspondance du modèle d'objet «Real Time Status» n'est utilisée que dans un «Information Report» MMS et renvoie un «MMS Access Result» OBJECT-ACCESS-DENIED en lecture et en écriture.

Les autres attributs d'objet correspondent comme suit:

AvailabilityReferenceID

Correspond au COMPOSANT **AvailabilityReferenceID** d'une Variable Nommée MMS de nom et de type sélectionnés.

Timestamp

Correspond au COMPOSANT **Timestamp** d'une Variable Nommée MMS de nom et de type sélectionnés.

PlantReferenceID

Correspond au COMPOSANT **PlantReferenceID** d'une Variable Nommée MMS de nom et de type sélectionnés.

UnitID

Correspond au COMPOSANT **UnitID** d'une Variable Nommée MMS de nom et de type sélectionnés.

Availability Status

L'attribut Availability Status sert à la sélection du nom et du type de la Variable Nommée MMS représentant le compte rendu.

MaxRampRateUp

S'il est présent, correspond au champ **MaxRampRateUp** du COMPOSANT **RampStatus** de la Variable Nommée MMS de type **StatusAvailable**.

MaxRampRateDown

S'il est présent, correspond au champ **MaxRampRateDown** du COMPOSANT **RampStatus** de la Variable Nommée MMS de type **StatusAvailable**.

Chacun des attributs suivants (quand il est présent) correspond à un champ du COMPOSANT **CapacityStatus** (type **UnitCapacity**) de la Variable Nommée MMS de type **StatusAvailable**.

UnitCapacity

S'il est présent, correspond au COMPOSANT **CapacityClass** de type **UnitCapacity** avec les significations suivantes: bit 0 = 1 implique GROSS, bit 1 = 1 implique NET, bit 0 et bit 1 = 1 implique BOTH.

GrossMaxCapacity

S'il est présent, correspond au COMPOSANT **GrossMaxCapacity** de type **UnitCapacity**. Si le bit 0 du COMPOSANT **CapacityClass** n'est pas = 1, **GrossMaxCapacity** doit être ignoré.

GrossMinCapacity

S'il est présent, correspond au COMPOSANT **GrossMinCapacity** de type **UnitCapacity**. Si le bit 0 du COMPOSANT **CapacityClass** n'est pas = 1, **GrossMinCapacity** doit être ignoré.

NetMaxCapacity

S'il est présent, correspond au COMPOSANT **NetMaxCapacity** de type **UnitCapacity**. Si le bit 1 du COMPOSANT **CapacityClass** n'est pas = 1, **NetMaxCapacity** doit être ignoré.

NetMinCapacity

S'il est présent, correspond au COMPOSANT **NetMinCapacity** de type **UnitCapacity**. Si le bit 1 du COMPOSANT **CapacityClass** n'est pas = 1, **NetMinCapacity** doit être ignoré.

Chacun des attributs suivants (quand il est présent) correspond à un champ dans le COMPOSANT **AvailabilityStatus** (type **AvailabilityClass**) de la Variable Nommée MMS de type **StatusAvailable**:

TypeOfAvailability

Correspond au bit 0 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=STANDBY, 1=ONLINE.

TimeToOnline

Correspond au COMPOSANT **TimeToOnline** de type **AvailabilityClass**. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 1, cet attribut doit être ignoré.

LFC

Correspond au bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES.

Dispatchable

Correspond au bit 2 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

Regulating

Correspond au bit 3 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

Manually_Loaded

Correspond au bit 4 du COMPOSANT **AvailFlags** de type **AvailabilityClass** avec les significations suivantes: 0=NO, 1=YES. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 0, cet attribut doit être ignoré.

ReasonForNoLFC

Correspond au COMPOSANT **ReasonForNoLFC** de type **AvailabilityClass** avec les significations suivantes: 0=STARTUP, 1=UNSTABLE. Si le bit 1 du COMPOSANT **AvailFlags** de type **AvailabilityClass** est 1, cet attribut doit être ignoré.

ReasonForUnavailable

Correspond au COMPOSANT **ReasonForUnavailable** de la Variable Nommée MMS de type **StatusUnAvailable** avec les significations suivantes: 0=FORCED, 1=SCHEDULED, 2=TESTING, 3=EQUIPMENT.

ProvidingReserve

Correspond au COMPOSANT **ProvidingReserve** de la Variable Nommée MMS sélectionnée (type **StatusAvailable** ou **StatusUnAvailable**) avec les significations suivantes: 0=NO, 1=YES.

C.3.3 Correspondance de Forecast

L'objet Forecast Report correspond à une Variable Nommée MMS appelée **Forecast** de type **Forecast**.

Les objets Forecast correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chacune des Variables Nommées MMS servant à la correspondance du modèle d'objet Forecast n'est utilisée que dans un «MMS Information Report» et renvoie un «MMS Access Result» OBJECT-ACCESS-DENIED en lecture et en écriture.

Les attributs de l'objet correspondent comme suit:

ForecastScheduleReferenceID

Correspond au COMPOSANT **ForecastScheduleReferenceID** d'une Variable Nommée MMS appelée **Forecast**.

PlantReferenceID

Correspond au COMPOSANT **PlantReferenceID** d'une Variable Nommée MMS appelée **Forecast**.

UnitID

Correspond au COMPOSANT **UnitID** d'une Variable Nommée MMS appelée **Forecast**.

ForecastType

Correspond au COMPOSANT **ForecastType** d'une Variable Nommée MMS appelée **Forecast** avec les significations suivantes: bit 0 à 1 pour GENERATION, bit 1 à 1 pour RESERVE, bit 0 et bit 1 à 1 pour BOTH.

StartTime

Correspond au COMPOSANT **StartTime** d'une Variable Nommée MMS appelée **Forecast**.

PeriodResolution

Correspond au COMPOSANT **PeriodResolution** d'une Variable Nommée MMS appelée **Forecast**.

NumberOfPeriods

Correspond au COMPOSANT **NumberOfPeriods** d'une Variable Nommée MMS appelée **Forecast**.

ListOfForecasts

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de ListOfForecasts:

- a) inscrire l'identificateur de valeur en virgule flottante **MW** comme la valeur d'une Variable Nommée MMS appelée **Matrix_Id** de type **MatrixId**;
- b) pour chaque période de temps transmise, inscrire la valeur MW comme la valeur d'une Variable Nommée MMS appelée **Float_Array_1** de type **FloatArray1**;
- c) inscrire l'identificateur de valeur entière **LFC_Code** comme la valeur d'une Variable Nommée MMS appelée **Matrix_Id** de type **MatrixId**;
- d) pour chaque période de temps, inscrire la valeur de LFC_Code comme la valeur d'une Variable Nommée MMS appelée **Integer_Array_1** de type **IntegerArray1**.

C.3.4 Correspondance de Curve

Les objets Curve correspondent à une suite de Variables Nommées MMS. La première variable (nom **Curve**, type **Curve**) définit les informations globales relatives à la courbe: nom, identificateurs de références, type de courbe et nombre de segments de la courbe. Chaque segment est représenté par:

- a) une Variable Nommée MMS (nom **Curve_Segment**, type **CurveSegmentDescription**);
- b) une Variable Nommée MMS (nom **Float_Array_XX**, type **FloatArrayXX**, où XX est le nombre de coefficients servant à décrire le segment de courbe).

Les objets Curve correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chaque Variable Nommée MMS utilisée pour la correspondance du modèle d'objet Curve ne sert que dans un «MMS Information Report» et renvoie un «MMS Access Result» OBJECT-ACCESS-DENIED en lecture et en écriture.

Les attributs de Curve correspondent comme suit:

CurveName

Correspond au COMPOSANT **CurveName** d'une Variable Nommée MMS de type **Curve**.

PlantReferenceID

Correspond au COMPOSANT **PlantReferenceID** d'une Variable Nommée MMS de type **Curve**.

UnitID

Correspond au COMPOSANT **UnitID** d'une Variable Nommée MMS de type **Curve**.

CurveType

Correspond au COMPOSANT **CurveType** d'une Variable Nommée MMS de type **Curve**.

NumberOfSegments

Correspond au COMPOSANT **NumberOfSegments** d'une Variable Nommée MMS de type **Curve**.

SequenceOfCurveSegmentDescription

Correspond à une Variable Nommée MMS appelée **Curve_Segment** de type **CurveSegmentDescription**. La correspondance des attributs du segment de courbe est comme suit:

Order

Correspond au COMPOSANT **Order** d'une Variable Nommée MMS de type **CurveSegmentDescription**.

LowRange

Correspond au COMPOSANT **LowRange** d'une Variable Nommée MMS de type **CurveSegmentDescription**.

HighRange

Correspond au COMPOSANT **HighRange** d'une Variable Nommée MMS de type **CurveSegmentDescription**.

SequenceOfCoefficients

Correspond aux éléments d'une Variable Nommée MMS de type **FloatArrayXX**, où XX est la valeur de l'attribut **Order**.

C.4 Correspondance de General Data Report

C.4.1 Correspondance de General Data Request

Les General Data Reports ne sont visibles qu'au travers du mécanisme Transfer Account Reporting (compte rendu de compte de transfert) comme défini dans l'IEC 60870-6-503. Les General Data Reports correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement instantanément, pendant que le compte rendu est généré. Chacune des Variables Nommées MMS utilisées pour la correspondance du modèle d'objet General Data Report n'est utilisée que dans les comptes rendus d'Information MMS et renvoie le Résultat d'Accès MMS OBJECT-ACCESS-DENIED (accès refusé) en lecture et en écriture.

Chacun des General Data Reports contient des informations d'en-tête et éventuellement une ou plusieurs listes de données (ListOfLocalReference, ListOfFloatingPoint1Values, ListOfFloatingPoint2Values, ListOfInteger1Values, ListOfInteger2Values, ListOfText1Values, ListOfText2Values). Les listes de données sont représentées comme suit:

ListOfLocalReference – suite de Variables Nommées MMS, chacune nommée **Reference_Num** et de type **ReferenceNum**. Il doit y avoir une Variable Nommée MMS pour chaque entrée de la ListOfLocalReference.

ListOfFloatingPoint1Values/ListOfFloatingPoint2Values – suite des Variables Nommées MMS ci-dessous. (A noter que chaque variable FloatArray ci-dessous représente un élément distinct de la ListOfFloatingPoint1Values ou de la ListOfFloatingPoint2Values).

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en virgule flottante inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données en virgule flottante;

Une Variable Nommée MMS **Float_Array_XX**, type **FloatArrayXX**.

ListOfInteger1Values/ListOfInteger2Values – suite des Variables Nommées MMS ci-dessous. (A noter que chaque variable FloatArray ci-dessous représente un élément distinct de la ListOfInteger1Values ou de la ListOfInteger2Values).

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en valeur entière inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données en valeur entière;

Une Variable Nommée MMS **Integer_Array_XX** de type **IntegerArrayXX**.

ListOfText1Values/ListOfText2Values – suite des Variables Nommées MMS ci-dessous. (A noter que chaque variable Text32Array ci-dessous représente un élément distinct de la ListOfText1Values ou de la the ListOfText2Values).

Une Variable Nommée MMS **Matrix_Id** de type **MatrixId** pour chaque grandeur en valeur texte inscrite dans le compte rendu. Cela définit la signification de chaque «colonne» de données en valeur texte;

Une Variable Nommée MMS **Text32_Array_XX**, type **Text32ArrayXX**.

L'ordre de transmission des Variables Nommées MMS représentant le GeneralDataReport doit être le suivant:

- a) Variable TAConditions_Detected (voir 7.1.4.4 de l'IEC 60870-6-503:2002)
- b) Variable Request_Id, si le compte rendu est une réponse à une Opération de Requête (voir 7.1.5.1.2 de l'IEC 60870-6-503:2002);
- c) La variable General_Data_Report de type GeneralDataReport représentant l'en-tête de GeneralDataReport
- d) Variables représentant la ListOfLocalReference, s'il y en a;
- e) Variables représentant la ListOfFloatingPoint1Values, s'il y en a;
- f) variables représentant la ListOfFloatingPoint2Values, s'il y en a;
- g) variables représentant la ListOfInteger1Values, s'il y en a;
- h) variables représentant la ListOfInteger2Values, s'il y en a;
- i) variables représentant la ListOfText1Values, s'il y en a;
- j) variables représentant la ListOfText2Values, s'il y en a;

La correspondance spécifique des attributs de l'objet General Data Report est la suivante:

GeneralDataReportReferenceNumber

Correspond au COMPOSANT **GeneralDataReportReferenceNumber** d'une variable MMS de type **GeneralDataReport**.

ReportName

Correspond au composant **ReportName** d'une Variable Nommée MMS de type **GeneralDataReport**.

ReportDateAndTime

Correspond au COMPOSANT **ReportDateAndTime** d'une variable MMS de type **GeneralDataReport**.

TransactionCode

Correspond au composant **TransactionCode** d'une Variable Nommée MMS de type **GeneralDataReport**.

NumberOfLocal Reference

Correspond au COMPOSANT **NumberOfLocalReference** d'une Variable Nommée MMS de type **GeneralDataReport**.

ListOfLocalReference

Correspond à une suite de Variables Nommées MMS appelées chacune **Reference_Num** et de type **ReferenceNum**. Il doit y avoir une référence de Variable Nommée MMS dans le compte rendu pour chaque entrée de la **ListOfLocalReference**. Le nombre de références **Reference_Num** du compte rendu doit être cohérent avec la valeur de l'attribut **NumberOfLocalReference**.

NumberOfFloatingPoint1

Correspond au COMPOSANT **NumberOfFloatingPoint1** d'une Variable MMS de type **GeneralDataReport**.

NumberOfFloatingPoint1Rows

Correspond au COMPOSANT **NumberOfFloatingPoint1Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfFloatingPoint1Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfFloatingPoint1Values**:

Inscrire les identificateurs des valeurs en virgule flottante en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Float_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

NumberOfFloatingPoint2

Correspond au COMPOSANT **NumberOfFloatingPoint2** d'une Variable MMS de type **GeneralDataReport**.

NumberOfFloatingPoint2Rows

Correspond au COMPOSANT **NumberOfFloatingPoint2Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfFloatingPoint2Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfFloatingPoint2Values**:

Inscrire les identificateurs des valeurs en virgule flottante en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Float_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

NumberOfInteger1

Correspond au COMPOSANT **NumberOfInteger1** d'une Variable MMS de type **GeneralDataReport**.

NumberOfInteger1Rows

Correspond au COMPOSANT **NumberOfInteger1Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfInteger1Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfInteger1Values**:

Inscrire les identificateurs des valeurs entières en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Integer_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

NumberOfInteger2

Correspond au COMPOSANT **NumberOfInteger2** d'une Variable MMS de type **GeneralDataReport**.

NumberOfInteger2Rows

Correspond au COMPOSANT **NumberOfInteger2Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfInteger2Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfInteger1Values**:

Inscrire les identificateurs des valeurs entières en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Integer_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

NumberOfText1

Correspond au COMPOSANT **NumberOfText1** d'une Variable Nommée MMS de type **GeneralDataReport**.

NumberOfText1Rows

Correspond au COMPOSANT **NumberOfText1Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfText1Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfText1Values**:

Inscrire les identificateurs des valeurs en texte en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Text32_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

NumberOfText2

Correspond au COMPOSANT **NumberOfText2** d'une Variable Nommée MMS de type **GeneralDataReport**.

NumberOfText2Rows

Correspond au COMPOSANT **NumberOfText2Rows** d'une Variable MMS de type **GeneralDataReport**.

ListOfText2Values

Représentée dans le compte rendu par une suite de Variables Nommées MMS. Pour chaque entrée de **ListOfText2Values**:

Inscrire les identificateurs des valeurs en texte en tant que valeurs d'une Variable Nommée MMS appelée **Matrix_Id** et de type **MatrixId**;

Pour chaque ligne, inscrire ces valeurs comme les valeurs d'une Variable Nommée MMS appelée **Text32_Array_XX**, où XX est assez grand pour contenir toutes les valeurs en virgule flottante.

C.4.2 Correspondance de General Data Response

L'objet General Data Response correspond à une Variable Nommée MMS appelée **General_Data_Response** de type **GeneralDataResponse**.

Les objets General Data Response correspondent à une suite d'une ou plusieurs Variables Nommées MMS, mais seulement de façon instantanée, pendant la génération du compte rendu. Chaque Variable Nommée MMS utilisée pour la correspondance du modèle d'objet General Data Response ne sert que dans un «MMS Information Report» et renvoie un «MMS Access Result» OBJECT-ACCESS-DENIED en lecture et en écriture.

Chaque objet General Data Response contient les informations d'en-tête et éventuellement la ListOfLocalReference, où:

ListOfLocalReference – suite de Variables Nommées MMS, chacune nommée **Reference_Num** et de type **ReferenceNum**. Il doit y avoir une Variable Nommée MMS pour chaque entrée de la **ListOfLocalReference**.

La correspondance spécifique des attributs de l'objet General Data Response est la suivante:

ReportReferenceNumber

Correspond au COMPOSANT **ReportReferenceNumber** d'une Variable MMS de type **GeneralDataResponse**.

ReportName

Correspond au COMPOSANT **ReportName** d'une Variable MMS de type **GeneralDataResponse**.

ReportTimeStamp

Correspond au COMPOSANT **ReportTimeStamp** d'une Variable MMS de type **GeneralDataResponse**.

NumberOfLocal Reference

Correspond au COMPOSANT **NumberOfLocalReference** d'une Variable MMS de type **GeneralDataResponse**.

ResponseData

Correspond au COMPOSANT **ResponseData** d'une variable MMS de type **GeneralDataResponse**.

ResponseCode

Correspond au COMPOSANT **ResponseCode** d'une Variable MMS de type **GeneralDataResponse**.

ResponseText

Correspond au COMPOSANT **ResponseText** d'une Variable MMS de type **GeneralDataResponse**.

Annexe D (informative)

Exemples de comptes de transfert

Dans ce scénario exemple, la compagnie A envoie à la compagnie E 10 000 MW pendant la première heure et 20 000 MW pendant la seconde heure. L'énergie est transmise à E via trois voies différentes (via la compagnie B, via la compagnie C et via la compagnie D). Il y a donc six segments de transmission:

Segment	Première heure (MW)	Seconde heure (MW)
A-B	5 000	10 000
B-E	5 000	10 000
A-C	2 000	4 000
C-E	2 000	4 000
A-D	3 000	6 000
D-E	3 000	6 000

Le total de l'énergie transférée est de 10 000 MW pendant la première heure et de 20 000 MW pendant la seconde. Pour chaque segment, on fournit une grandeur en virgule flottante (par exemple MW). Pour la transaction complète, on fournit deux grandeurs en virgule flottante (par exemple le prix et la quantité totale de MW).

Cette transaction doit être représentée comme suit:

- a) La première Variable Nommée MMS décrit les conditions qui ont provoqué le compte rendu:

```
nom TA_Conditions_Detected, type TAConditionsDetected, valeur:
    bitstring, with bit 0 set (BeforeTheHour)
```

- b) La Variable Nommée MMS suivante est de type TASegmentsPeriodic, appelée TA_Segments_Periodic, et de valeur:

```
TA_Segments_Periodic
{
    TransferAccountRef:    1 002
    SendUtility:          A
    RecvUtility:          E
    SellingUtility:       A
    BuyingUtility:        E
    TimeStamp:            <heure de transmission>
    Name:                  <nom>
    StartTime:            12:00
    PeriodResolution      3 600 s
    NumberLocalReferences: 0
    NumberOfSegments:    6
    NumberFloatIds       2
    NumberIntegerIds     0
    NumberPeriods        2
}
```

- c) La variable suivante est de type TATransmissionSegment, son nom est TA_Transmission_Segment, représentant la variable d'en-tête pour le segment:

TA_Transmission_Segment

```
{
    TransmissionReference      (A-B)
    UtilWheeling:              B
    UtilPaying:                 E
    TransmissionSegType:       OUT_ONLY
    UtilIn:                     A
    UtilOut:                    B
    InterchangePtIn:           <ignoré>
    InterchangePtOut:          <ignoré>
    InterchangePt:              < référence locale du lien >
    NumberFloatIds              1
    NumberIntegerIds           0
}
```

- d) La variable suivante rend compte de la signification de ce dont on rend compte pour le segment. Etant donné qu'une seule valeur est fournie (une colonne dans la matrice conceptuelle), il n'y a qu'un seul identificateur.

Matrix_Id, type MatrixId, valeur: MW

- e) Etant donné qu'il y a deux périodes de temps (heures) pour le compte rendu de compte, il y a deux variables (lignes de la matrice conceptuelle) transmises, chacune de longueur 1:

Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] = 5 000)

Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] = 10 000)

- f) La suite (TA_Transmission_Segment, Matrix_Id et Float_Array1) est répétée pour chacun des autres segments objets du compte rendu.

Pour le segment 2:

TA_Transmission_Segment

```
{
    TransmissionReference      (B-E)
    ...
    NumberFloatIds            1
    NumberIntegerIds          0
}
```

Matrix_Id, type MatrixId, valeur: MW

Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] = 5 000)

Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] = 10 000)

Pour le segment 3:

TA_Transmission_Segment

```
{
    TransmissionReference      (A-C)
    ...
    NumberFloatIds              1
```

```

        NumberIntegerIds          0
    }
    Matrix_Id                      MW
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    2 000)
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    4 000)

```

Pour le segment 4:

```

    TA_Transmission_Segment
    {
        TransmissionReference      (C-E)
        ...
        NumberFloatIds            1
        NumberIntegerIds          0
    }
    Matrix_Id                      MW
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    2 000)
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    4 000)

```

Pour le segment 5:

```

    TA_Transmission_Segment
    {
        TransmissionReference      (A-D)
        ...
        NumberFloatIds            1
        NumberIntegerIds          0
    }
    Matrix_Id                      MW
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    3 000)
    Float_Array1, type FloatArray1, valeur représentant le total des MW (Float_Array1[0] =
    6 000)

```

Pour le segment 6:

```

    TA_Transmission_Segment
    {
        TransmissionReference      (D-E)
        ...
        NumberFloatIds            1
        NumberIntegerIds          0
    }
    Matrix_Id                      MW
    Nom: Float_Array1, type FloatArray1, valeur représentant le total des MW
    (Float_Array1[0] = 3 000)

```

Nom: Float_Array1, type FloatArray1, valeur représentant le total des MW
(Float_Array1[0] = 6 000)

Pour terminer, les variables représentant la transaction complète entre A et E peuvent être incluses:

Matrix_Id	PRICE
Matrix_Id	MW

Nom: Float_Array1, type FloatArray1, valeurs pour le prix et le total des MW de la 1ère heure:
Float_Array1[0] = 10 000, Float_Array1[1] = 50,00

Nom: Float_Array1, type FloatArray1, valeurs pour le prix et le total des MW de la 2ème
heure:
Float_Array1[0] = 20 000, Float_Array1[1] = 50,00

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch