TECHNICAL REPORT

IEC TR 60870-6-505

First edition
2002-08

Telecontrol equipment and systems –

Part 6-505:
Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 User guide

Reference number
IEC/TR 60870-6-505:2002(E)

Publication numbering

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

Consolidated editions

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

Further information on IEC publications

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

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

- **Catalogue of IEC publications**

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 On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

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

- **Customer Service Centre**

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

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

TECHNICAL REPORT

IEC TR 60870-6-505

First edition
2002-08

Telecontrol equipment and systems –

Part 6-505:
Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 User guide

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

Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE XA

For price, see current catalogue

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

TELECONTROL EQUIPMENT AND SYSTEMS –

Part 6-505: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 User guide

FOREWORD

1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the 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, the IEC publishes International Standards. 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. The 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 the 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 National Committees.
3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
6) Attention is drawn to the possibility that some of the elements of this technical report may be the subject of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example “state of the art”.

Technical reports do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful by the maintenance team.

IEC 60870-6-505, which is a technical report, has been prepared by IEC technical committee 57: Power system control and associated communications.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
57/548/CDV	57/580/RVC

Full information on the voting for the approval of this technical report 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.

This document, which is purely informative, is not to be regarded as an International Standard.

The committee has decided that the contents of this publication will remain unchanged until 2007. At this date, the publication will be

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

INTRODUCTION

A large number of utilities use the Telecontrol Application Service Element.2 (TASE.2), an international standard protocol for communication of real-time data. TASE.2 provides a common means for all utilities to exchange data between not only control centers, but power plants and substations as well. The adoption of TASE.2 has lead to the availability of competitively priced data communication products based on TASE.2 from multiple vendors at a fraction of the cost of a proprietary system. This report provides guidance for utility users who are evaluating, procuring, and configuring TASE.2, as well as aid to vendors implementing TASE.2 in their products. The individual server and data objects comprising TASE.2 are described, with cross references to the specification. This provides the reader the basic understanding needed to use the TASE.2 specifications in an informed manner. The guide then addresses practical issues that arise in connection with TASE.2 use.

TELECONTROL EQUIPMENT AND SYSTEMS –

Part 6-505: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 User guide

1 Scope

This technical report provides a set of guidelines on the use of the following TASE.2 international standards:

IEC 60870-6-503

IEC 60870-6-702

IEC 60870-6-802

These standards specify a method of exchanging time-critical control center data through wide- and local-area networks using a full ISO compliant protocol stack. They contain provisions for supporting both centralized and distributed architectures. These standards include the exchange of real-time indications, control operations, time series data, scheduling and accounting information, unstructured ASCII or binary files, remote program control, and event notification.

However, the style of the TASE.2 standards may make them somewhat difficult to read for someone either not familiar with the precise syntax of the language used to describe the protocol or with all the background leading up to the development of these specifications. Furthermore, certain types of information that may be very useful to a user of TASE.2 but not necessary for specifying the protocol or services provided by TASE.2 have been omitted. Thus the need for this User Guide.

1.1 Intended users

This User Guide is intended for a broad audience of readers from an end user trying to decide if TASE.2 is appropriate for their data transfer needs to a vendor planning to implement TASE.2, with the goal of offering a TASE.2 product. In particular, this guide should be helpful to the following:

- An end user, such as an electric utility, with the need to transfer real-time data to another utility or utilities or to another internal control center, who is trying to evaluate which protocol is the most appropriate.
- An end user who has already decided to use TASE.2 and now needs guidance in how to procure TASE.2.
- An end user who has procured TASE.2 and is now concerned about how to map their actual data into TASE.2 data objects exactly.
- An end user who is looking for conventions and answers to practical questions regarding configuring TASE.2 software and networks.
- A vendor with a project to implement the TASE.2 specification either as a special project or to offer a standard product.

1.2 Organization

This guide first introduces the background and concepts of TASE.2 to provide a framework for understanding the TASE.2 specification. Then the individual server and data objects comprising TASE.2 are described with cross references into the specification. At this point, (i.e., Clauses 1-8) the reader should have all the necessary basic understanding to use the TASE.2 specifications intelligently. The remainder of the guide (Clauses 9-20) address practical issues that arise in connection with the use of TASE.2.

1.3 TASE.2 Version

This edition of the TASE.2 User Guide was prepared using the Second Edition of the TASE.2 standards, which at the time of the preparation of this report was the 2000 edition.

2 Reference documents

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

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

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

IEC 60870-6-702:1998, *Telecontrol equipment and systems – Part 6-702: Telecontrol protocols compatible which ISO standards and ITU-T recommendations – Functional profile for providing the TASE.2 application service in end systems*

IEC 60870-6-802:2002, *Telecontrol equipment and systems – Part 6-802: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Object models*

ISO/IEC 8802-2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical link control*

ISO/IEC 9506 (all parts), *Industrial automation systems – Manufacturing message specification*

3 Definitions

For the purposes of this technical report, the following definitions apply.

3.1
action
activity performed by the TASE.2 server triggered by some local predefined stimulus or set of circumstances

3.2
accounting information
set of information which describes an account for a utility. See IEC 60870-6-802 for more details

3.3
bilateral agreement
agreement between two control centers which identifies the data elements and objects that can be accessed and the level of access permitted

3.4
bilateral table
computer representation of the Bilateral Agreement. The representation used is a local matter

3.5
client
TASE.2 user who requests services or objects owned by another TASE.2 user acting as a server. The client is a communicating entity which makes use of the VCC for the lifetime of an association via one or more TASE.2 service requests

3.6
data set
object which provides services to group data values for singular operations by a TASE.2 client

3.7
data value
object which represents some alphanumeric quantity that is part of the Virtual Control Center (VCC) which is visible to a TASE.2 user. Data Values exist as part of the implementation of the control center and represent either real entities within the utility such as current, or derived values calculated in the control center. Data Value objects include services for accessing and managing them

3.8
instance
implementation of TASE.2 executed in either the client or the server role

3.9
interchange schedule
set of information that specifies how energy is transferred from one system to another. See IEC 60870-6-802 for more details

3.10
object
abstract entity used to implement the TASE.2 protocol and to represent data and optionally provide services for accessing that data within a VCC

3.11
object model
abstract representation that is used for real data, devices, operator stations, programs, event conditions, and event enrollments

3.12
operation
activity which is performed by the TASE.2 server at the request of the TASE.2 client

3.13
server
TASE.2 user that is the source of data and provides services for accessing that data. A TASE.2 server behaves as a VCC over the lifetime of an association

3.14
service
activity which is either a TASE.2 action or operation

3.15
tagged
the term is derived from the practice of putting a physical tag on a device as it is turned off for servicing or locked out from network access as a safety measure. The TASE.2 term tagged is used to signal such a condition to the TASE.2 user

3.16
time series
set of values of a given element that is taken at different times as specified by a single time interval. A time series is implemented through the transfer set mechanism as defined within this specification

3.17
transfer account
set of information that associates interchange scheduling information with either hourly or profile data

3.18
transfer conditions
events or circumstances under which a TASE.2 server reports the values of a data set, values in a time series, or all transfer account information

3.19
transfer set
object used to control data exchange by associating data values with transmission parameters such as time intervals, for example. There are four types of Transfer Sets: Data Set Transfer Sets, Time Series Transfer Sets, Transfer Account Transfer Sets, and information Message Transfer Sets

3.20
user
implementation of TASE.2 executed in either the client or the server role

3.21
Virtual Control Center (VCC)
abstract representation of a real control center which describes a set of behaviour with regards to communication and data management functionality and limitations. VCC is a concept taken from the underlying MMS services

4 Abbreviations

For the purpose of this standard, the following abbreviations apply.

ACSE	Association Control Service Element
API	Application Program Interface
BCD	Binary Coded Decimal
COV	Change Of Value
DIS	Draft International Standard
EPRI	Electric Power Research Institute
HLO	Hot line order
ICC	Inter-Control Center
ICCP	Inter-Control Center Communications Protocol
IDEC	Inter-utility Data Exchange Consortium
IP	Internet Protocol

KQH Kilovar hour readings
KWH Kilowatt hour readings
LFC Load Following
MMS Manufacturing Messaging Specification
MOD Motor operated disconnect
PDU Protocol Data Unit
QOS Quality of Service
RBE Report by Exception
ROSE Remote Operations Service Element
TAL Time Allowed to Live
TASE Tele-control Application Service Element, (IEC's designation of an international standard protocol for utility data exchange)
TASE.1 TASE based on the ELCOM-90 protocol
TASE.2 TASE based on the ICCP protocol
TCP Transmission Control Protocol
TLE Time Limit for Execution
TOD Time of Day
UCA Utility Communications Architecture
UCS Utility Communications Standards working group
UDP User Datagram Protocol
VCC Virtual Control Center
VMD Virtual Manufacturing Device
WSCC Western System Coordinating Council
WEICG WSCC Energy Management Systems Inter-utility Communications Guidelines

5 TASE.2 background

Inter-utility real-time data exchange has become critical to the operation of inter-connected systems within the electric power utility industry. The ability to exchange power system data with boundary control areas and beyond provides visibility for disturbance detection and reconstruction, improved modelling capability and enhanced operation through future security control centers or independent system operators.

Historically, utilities have relied on in-house or proprietary, non-IS standard protocols such as those developed by the Western Systems Coordinating Council (WSCC), ELCOM, and the Inter-utility Data Exchange Consortium (IDEC) to exchange real-time data. TASE.2 began as an effort by power utilities, several major data exchange protocol support groups for the protocols mentioned above, EPRI, consultants and a number of SCADA/EMS and protocol vendors to develop a comprehensive, international standard for real-time data exchange within the electric power utilities industry.

By giving all interested parties an opportunity to provide requirements input and to participate in the protocol definition process, it was expected that the final product would both meet the needs of and be accepted by the electric power utility industry. To accomplish this goal, the Utility Communications Specification (UCS) Working Group was formed in September 1991 to:

1) develop the protocol specification;
2) develop a prototype implementation to test the specification;
3) submit the specification for standardization;
4) perform inter-operability tests among the developing vendors.

UCS submitted TASE.2 to the IEC Technical Committee (TC) 57 Working Group (WG) 07 as a proposed protocol standard. Another proposed standard based on ELCOM-90 over ROSE was also being considered by WG 07. TC 57 decided on a multi-standard approach to allow (1) a quick implementation to meet European Common Market requirements by 1992 and (2) also allow long term development of a more comprehensive protocol. The first protocol was designated TASE.1 (Telecontrol Application Service Element-1). The second protocol, based on TASE.2 over MMS, was designated TASE.2.

Successful first implementations of TASE.2 between SCADA/EMS control centers led to further expansion to allow communications between control centers and power plants. This expansion did not impact the basic services, but did lead to the development of specific power plant objects. These objects were incorporated into TASE.2. Similarly, protection event data objects were also added to support substation communications. The second edition of the TASE.2 standards also includes one new object for the exchange of general data reports and another new object for sending acknowledgements of complex data objects.

6 TASE.2 Overview

6.1 TASE.2 Concepts

6.1.1 Protocol Architecture

TASE.2 maximizes the use of existing standard protocols in all layers up to and including the lower layers of layer 7 in the OSI reference model. This has the benefit of requiring new protocol development for TASE.2 only in the upper sublayer of layer 7.

The protocol stack used by TASE.2 is shown in Figure 1. The upper three layers conform to the standard 7-layer OSI protocol stack with control center applications at the top. TASE.2 specifies the use of the Manufacturing Messaging Specification (MMS) for the messaging services needed by TASE.2 in layer 7. MMS specifies the mechanics of naming, listing, and addressing variables, and of message control and interpretation, while TASE.2 specifies such things as the control center object formats and methods for data requests and reporting. Applications at different control centers, possibly written by different vendors, but both conforming to these mechanics, formats, and methods, may interoperate to share data, control utility devices, output information messages, or define and execute remote programs.

Application	ICCP IEC 60870-6-503/802 MMS ACSE	
Presentation	ISO Presentation	
Session	ISO Session	
Transport	ISO Transport Class 4	TCP
Network	ISO CLNP	IP
Data Link	ISO 8802-2 LLC, FDDI, FR, ISDN, etc.	
Physical	Ethernet LAN, WAN, Point-to-Point Circuit, ATM, SDH, etc.	

Figure 1 – TASE.2 Protocol Architecture

TASE.2 also utilizes the services of the Application Control Service Element (ACSE) in layer 7 to establish and manage logical associations or connections between sites. TASE.2 relies on the ISO Presentation Layer 6 and Session Layer 5 as well.

Because of the protocol architecture, TASE.2 is independent of the lower layers, so that as new protocols evolve in the lower layers, TASE.2 will be able to operate over them with only configuration changes. Thus TASE.2 is able to operate over either an ISO-compliant transport layer or a TCP/IP transport service, as long as ISO layers 5-7 are maintained.

IEC 60870-6-702, which is a Protocol Conformance Protocol Specification (PICS), provides details on the requirements of each of the protocols in Layers 5-7. This information is especially important to protocol stack providers.

6.1.2 Application Program Interface (API)

An Application Program Interface (API) is not specified in the TASE.2 specification – only the protocol and service definitions are specified and are the subject of standardization. Each vendor implementing TASE.2 is free to choose the API most suitable for their product or for their intended customers. Figure 2 illustrates this concept.

For example, an Energy Management System/Supervisory Control And Data Acquisition (EMS/SCADA) vendor may choose to provide an API optimized for interfacing with several different types of applications, such as:

– A proprietary real-time SCADA database for the storing and retrieving of real-time power system data, such as analogs, status, and accumulator values, on a periodic basis or when a value changes

– A Relational Data Base Management System (RDBMS) for the storing and retrieving of historical or other non-real time data, or of Block 8 transfer account or device outage data objects.

– Scheduling and accounting applications to send, for example (1) interchange schedules once an hour or once a day and (2) binary files containing accounting data spreadsheet files.

– Dispatcher console operator message application and/or alarm processor application to send ASCII text messages to be displayed on a dispatcher's console display at another control center

These are just a few examples of the types of APIs an EMS/SCADA vendor may provide for its TASE.2 product. How they are implemented is considered a "local implementation issue" in the TASE.2 specification. As long as the protocol services are implemented according to the specification, interoperability is assured between different TASE.2 vendor's products.

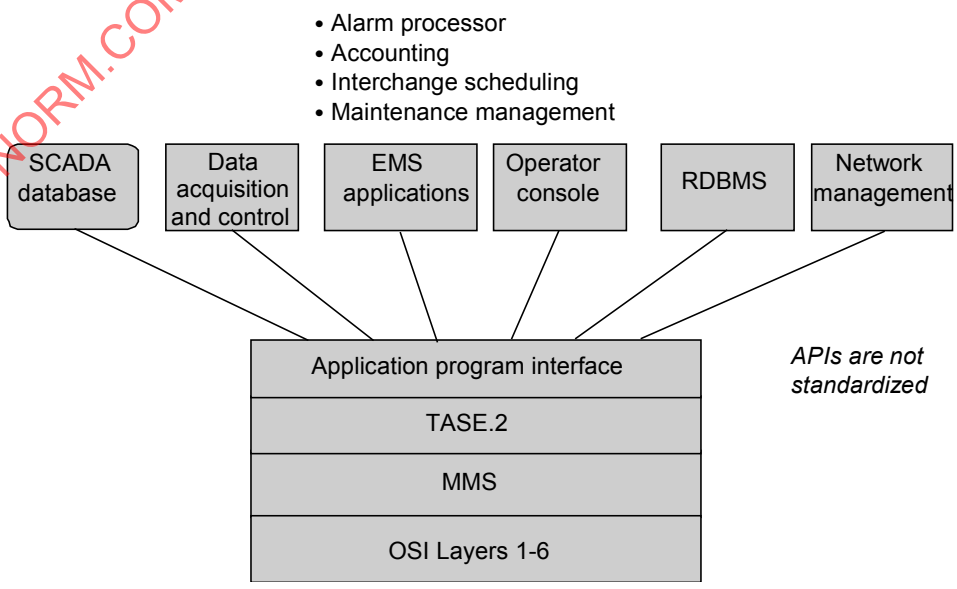

Figure 2 – Application Program Interface (API)

6.1.3 Client/Server Model

TASE.2 is based on client/server concepts. All data transfers originate with a request from a control center (the client) to another control center which owns and manages the data (the server). For example, if a Control Center X application needs data from the Control Center Y SCADA database, the Control Center X application acting as the client may request Control Center Y acting as the server to send the data under conditions specified by the client.

There are various services provided in TASE.2 to accomplish data transfers, depending on the type of request. For example, if the client makes a one-shot request, the data will be returned as a response to the request. However, if the client makes a request for the periodic transfer of data or the transfer of data only when it changes, then the client will first establish the reporting mechanism with the server (i.e., specify reporting conditions such as periodicity for periodic transfers or other trigger conditions such as report-by-exception only), and the server will then send the data as an unsolicited *report* whenever the reporting conditions are satisfied.

A control center may function as both a client and a server.

6.1.4 Multiple Associations and Sites

TASE.2 uses the ISO ACSE to establish logical associations. Multiple associations may be established from a client to multiple, different control center servers. Although TASE.2 may be operated over a point-to-point link, it is envisioned that most installations will operate over a router-based Wide Area Network (WAN). As noted previously, TASE.2 is independent of the underlying transport network, so any combination of subnetworks may comprise the WAN, including the LANs within a site.

Multiple associations may also be established to the *same* control center for the purpose of providing associations with different Quality Of Service (QOS). A TASE.2 client then uses the association with the appropriate QOS for the operation to be performed. For example, to ensure the real time data messages are not delayed by non-real data transfers, both a High and Low priority association may be established, with a separate message queue for each. TASE.2 will check the High priority message queue and service any messages queued before servicing the Low priority message queue. This permits a common data link to be shared for the transfer of both high priority SCADA data and lower priority information message transfers.

Figure 3 illustrates a TASE.2 network serving four utilities. As shown, Utility A is a client to server C (Association C1) and a server for four associations: two to client C (Association A1 and A2), one to client B (Association A3), and one to client D (Association A4). The association to client B (A3) would presumably be accomplished via a router at utility C, but could follow any path available if a WAN is provided to interconnect all utilities. Each of the other utilities shown have similar associations established to meet their individual needs. Utility D functions only as a client. Utilities B and C function as both clients and servers. The point made by this diagram is that TASE.2 provides the capability for any type of interconnectivity needed via configuration of the TASE.2 software.

Only a client is allowed to initially establish an association in TASE.2. When the TASE.2 users at each end of an association are capable of being both a client and a server, then the side to initiate the association is a system configuration issue.

TASE.2 optionally *permits* an established association to be used by either a client or server application at a site, independent of how the association was established. How associations are actually started and used in any actual configuration of TASE.2 is a system configuration issue for each particular installation. However, if a site that can utilise one association for both client-server directions (i.e., dual use) attempts to establish an association with a site that does not support dual use, it is the responsibility of the dual use site to fall back to single use associations.

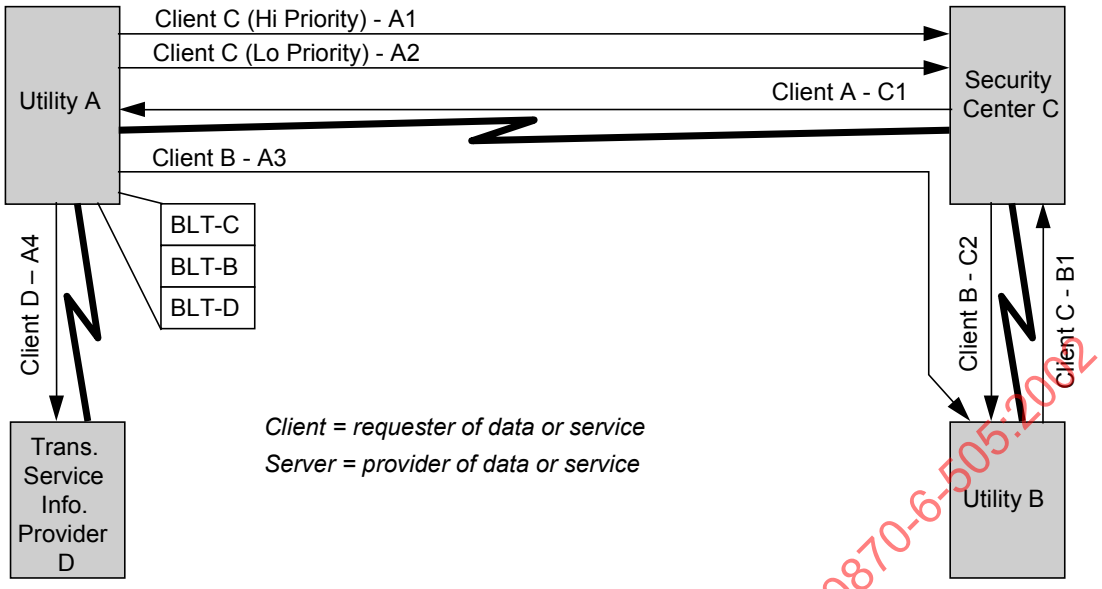

Figure 3 – TASE.2 Client/Server Model with Multiple Associations

6.1.5 Access Control via Bilateral Tables

To provide access control, the server checks each client request to ensure that that particular client has access rights to the data or capability requested. Access control is provided through the use of Bilateral Tables (BLTs) defined for each TASE.2 client that has one or more associations with a TASE.2 server. BLTs provide execute, read/write, read only, or no access for each item (which includes data values, devices, and programs) that can be requested by a client.

A BLT is the computer version of a bilateral agreement established by two control centers. Each control center acting as a *server* specifies for each control center acting as a *client* which objects it is willing to make available, and for each available object, which types of access are supported. It is only necessary for the server to maintain the BLT on-line, which serves as a kind of “shopping list” of available data objects from which the client can choose using the TASE.2 protocol services. For example, as shown in Figure 3, Utility A acting as a server maintains a separate BLT for each client utility (for example, Utilities B-D), providing different access rights for each client utility.

Once an association is established, before any other operations are attempted, the TASE.2 client reads the BLT version attribute and the TASE.2 version attribute maintained at the server. The client then compares the returned values with the values of its versions, and if there is a mismatch with either attribute, the client is required to conclude the association until the mismatches are resolved. Additionally, for any operations used by the TASE.2 client on any data objects after the association has been established, if the TASE.2 server is using an access control mechanism, it must perform the general validity check to ensure that all identifiers referenced in the operation are valid and covered in the Bilateral Table.

If inconsistencies are detected in the bilateral tables applicable to a particular association, the TASE.2 client may use the Get Data Value Names operation described in Subclause 7.2 below to obtain a list of the names of all the Data Value objects which it can access (as defined in the BLT version at the TASE.2 server). These can then be compared to the list of objects that the client believes should be accessible at that server. Any observed differences should be addressed by the network administrator and resolved to obtain identical BLT records at both the client and server.

6.1.6 Use of Object Models

Object model concepts are used in two different ways in TASE.2. Figure 4 illustrates these concepts.

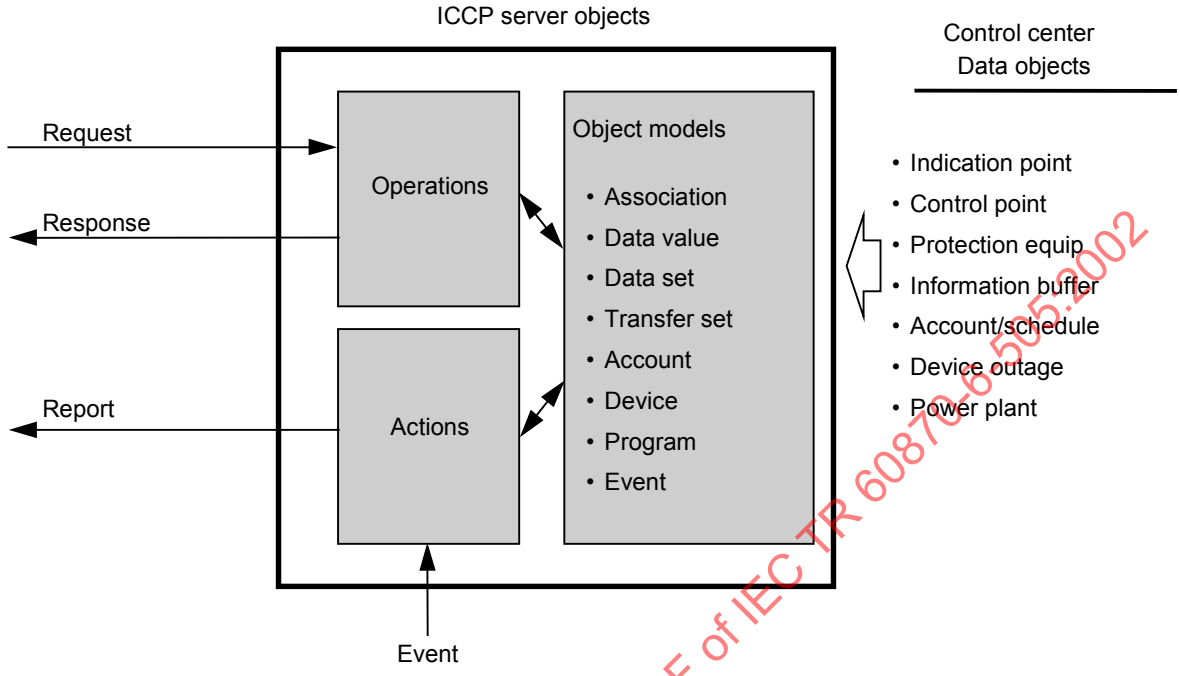

Figure 4 – TASE.2 Object Models

6.1.6.1 TASE.2 Server Objects

Firstly, all TASE.2 services are provided via TASE.2 server objects which may be thought of as classical objects with data attributes and methods as defined in object-oriented design methodologies. There are two basic types of methods in TASE.2 called *operations* and *actions*. An *operation* is client-initiated via a *request* to a server, typically followed by a *response* from the server. An *action*, on the other hand, is a server-initiated function. An example of an *action* is the transfer of data via a *report* to a client in response to a timer expiring or some other external event at the server, such as a change in status of a breaker.

IEC 60870-6-503 contains all the TASE.2 server object definitions. These objects are required to implement the TASE.2 protocol and are sometimes referred to as “internal” objects. Explanations of these objects is included in this guide in the TASE.2 Server Object Description subclause.

6.1.6.2 TASE.2 Data Objects

Secondly, all other data and control elements typically exchanged between control centers are defined as “data objects”. These range from simple to complex data structures. In contrast to the “server objects”, these objects are not required to implement the TASE.2 protocol, and so are sometimes referred to as “external” objects.

The standard Control Center Data Objects are defined in IEC 60870-6-802. They are also described in this guide in the Conformance Block clause. Supported data types include control messages, status, analogs, quality codes, schedules, text and simple files. Furthermore, additional data objects can be defined by TASE.2 users and transferred using existing TASE.2 server objects with no change in the TASE.2 protocol software contained in IEC 60870-6-503. The approach to defining new data objects is described in this guide in Clause 10.

6.1.6.3 Object Model Notation

The TASE.2 specification uses a formal method of describing objects. The first level is known as an Abstract Object Model. This model comprises a Name for the model, followed by a list of Attributes, headed by one Attribute known as a Key Attribute. In some cases, an Attribute listed actually another object model inherited by the new object model. The meaning of each attribute is provided after the formal object model is presented.

Some object models, especially those used to describe control center data objects, contain Constraints, which provide alternative lists of Attributes within a single object model. These Constraints thus provide some flexibility in how the object can be used. All abstract models are described first in the specification.

Abstract object models then have to be mapped to concrete Structures with Components. Each Component is mapped to a data type. The services are mapped to MMS services. This must be specified to ensure that each implementer of TASE.2 uses the same data types and MMS services to implement the abstract models so that interoperability can be achieved with other vendor's TASE.2 products.

Clause 7 describes in more detail the organization of the TASE.2 specification and the use of these models.

6.1.7 Conformance Blocks and Services

Conformance blocks are defined for TASE.2 for server objects in Clause 9 of IEC 60870-6-503 as a way of grouping TASE.2 objects together to provide fundamental types of services. A vendor need not implement all defined conformance blocks (nine in all). However, any implementation claiming conformance to TASE.2 must fully support Block 1, as defined in Clause 8 of this guide. Likewise, a TASE.2 end user need not procure all TASE.2 conformance blocks, only the ones actually needed to meet the user's data transfer requirements.

Conformance blocks are also defined in Clause 9 of IEC 60870-6-802 for data objects as a way of specifying which server object services are needed to transfer each data object defined in IEC 60870-6-802.

6.2 TASE.2 Specification Organization

The TASE.2 specification organization is dictated by the rules and guidelines governing IEC/ISO standards documentation. The IEC numbers for the three parts of the specification were assigned by the IEC, basically by assigning the next sequential numbers available in the IEC 60870-6-500, -700, and -800 series of documents. The 500 series numbers are reserved for protocol standards and service specifications. The 700 series is reserved for Application profiles. The 800 series is reserved for Information Structure profiles, also known as Interchange Format and Representation profiles. This follows the classification scheme adopted for OSI functional profiles.

This subclause should explain how the IEC documents are organized and why (i.e., separation into parts 503, 702, and 802).

6.2.1 IEC 60870-6-503

TASE.2 Part 503, known officially as IEC 60870-6-503, TASE.2 Services and Protocol, defines the mechanism for exchanging time-critical data between control centers. The data exchange mechanism is defined in terms of TASE.2 server object models. It defines a standard way of using the ISO/IEC 9506 MMS services to implement the data exchanges.

A document that defines a standard way to use selected MMS services for exchanging electric utility data is known as an MMS Companion Standard for Electric Utility Data Exchange. And since MMS Companion Standards must follow a consistent format dictated by the MMS standards development groups, IEC 60870-6-503 is formatted as it is largely to conform to the guidelines established for MMS Companion Standards. This means that readability is sometimes sacrificed to follow these guidelines.

The ordering of the information presented in the document follows the “onion skin” analogy. That is, reading this document is like peeling off multiple layers of onion skins, with each new layer taking the reader to a deeper level of specification. This means that the same models are discussed at different levels several times throughout the specification. The order is as follows:

Layer 1

Subclause 5.1: Informal TASE.2 Model Description. The informal model of TASE.2 describes the various TASE.2 server objects in the context of the utility control center environment using plain English narrative.

Layer 2

Subclause 5.2: Formal TASE.2 Model Description. The formal model covers the same ground, only here more formality is introduced. Specifically, the entire control center with its software applications that are involved in data exchange are represented as a Virtual Control Center (VCC), comprising several object models. In this subclause, formal abstract models with attributes are introduced. Some models are represented as a hierarchy of object models, each of which is described. Each attribute for each object model is defined. Each operation and action is described again in more detail.

Layer 3

This layer covers three major clauses:

Clause 6: Mapping of TASE.2 Object Models onto MMS Object Models. In this clause, the abstract object models are repeated, only this time each attribute is mapped directly to either a basic MMS attribute type or to a more complex TASE.2 data type defined in Clause 8, so that standard MMS protocols can be used for the actual transmission of data. For example, in Subclause 5.2 the Data Set Name attribute of the Data Set object model is defined as “the attribute that uniquely identifies the Data Set.” In Clause 6, the description for the Data Set Name attribute states that “this attribute shall be represented as the MMS Variable List Name attribute.”

Clause 7: Mapping of TASE.2 Operations and Action onto MMS Services. This clause does for operations and actions what Clause 6 does for attributes. It maps them onto MMS services, describing both the client and server roles in sufficient detail so that a software vendor can implement each service in such a way that interoperability with other vendor’s TASE.2 products is assured.

Many of the TASE.2 operations use the MMS Read and Write services. These services have the ability to read or write, respectively, to one or more MMS variables at a time. A positive MMS Read or Write service response is sent back to the client from the server if the MMS service request was acceptable to the VMD, and additionally:

a) if it is an MMS Read service request, the VMD attempted to determine the value of each of the MMS variables in the request; or

b) if it is an MMS Write service request, the VMD attempted to replace the value of each of the specified MMS variables with the value supplied in the request.

Because of this, a positive MMS service response could be sent back from the server but indicate a failure on reading or writing to any or all of the MMS variables. This is true even when only one MMS variable is indicated in the MMS Read or Write service request.

Clause 8: Standardized Application-Specific Objects. This clause specifies certain TASE.2 objects and complex data types used in Clause 6 and maps them onto MMS standard objects and basic MMS data types. This clause deals only with the objects required for TASE.2 internal use as distinguished from control center data objects, which are the subject of IEC 60870-6-802.

The last part of IEC 60870-6-503, Clause 9, defines the Conformance Blocks, which are described elsewhere in this guide.

6.2.2 IEC 60870-6-802

TASE.2 Part 802, known officially as IEC 60870-6-802, TASE.2 Object Models, defines the control center data objects, which represent the control center data actually exchanged between control centers. This document is structured in a fashion somewhat similar to IEC 60870-6-503, but with only two layers:

Layer 1

Clause 5: Object Models. This clause defines the standard abstract object models for the data to be exchanged with TASE.2. This uses the same notation that was used in IEC 60870-6-503 to describe the TASE.2 server object models, defining each attribute for each model. This is the clause to browse or read to determine if there are appropriate standard objects available to meet a specific utility data exchange requirements. It is organized based on the classes of data typically exchanged between control centers. The order is Supervisory Control and Data Acquisition, Transfer Accounts, Device Outage, Information Buffer, and Power Plant objects.

Layer 2

This layer comprises two clauses.

Clause 6: MMS Types for Object Exchange. This clause defines the data types to be used for exchanging the standard objects. This includes basic types, such as Data_Discrete, which is defined as an integer {width 32}. But it also includes complex data types based on the abstract models defined in Clause 5. Each abstract model must be mapped to one or more concrete object types, which are defined in terms of structures with components. For example, an Indication Point object which contains an analog point value with quality and time tag (but no change-of-value counter) is mapped to the Data_DiscreteQTimeTag type, which is a complex type with a structure containing the components Value, TimeStamp, and Flags. Each component is also mapped to a data type, in this case Data_Discrete, Data_TimeStamp, and Data_Flags, respectively. In all cases, each type maps down to a supported MMS type for data exchange.

This clause contains both the basic types and complex structure types, ordered the same as Clause 5. The exception is the Matrix Data Type, which is used by several different objects, as described in Clause 7.

Clause 7: Mapping of Object Models to MMS Types. These clauses defines the mapping of each object attribute from Clause 5 to one or more of the TASE.2 types defined in Clause 6.

Clause 8, Use of Supervisory Control Objects, provides examples in the use of the Supervisory Control objects in order to introduce some conventions in assigning meaning to certain attributes which are generic in nature.

Clause 9, Conformance, identifies the IEC 60870-6-503 Conformance Block required to provide the necessary services for exchanging each data object described in IEC 60870-6-802.

6.2.3 IEC 60870-6-702

This document is a Protocol Implementation Conformance Specification (PICS) which defines the Application Profile (Layers 5-7) for use with TASE.2. It is needed for vendors implementing protocol stacks that support the TASE.2 application layer. Most users of TASE.2 will not be concerned with this specification. Therefore this guide does not deal specifically with IEC 60870-6-702.

7 TASE.2 Server Objects

7.1 Association

Association objects are used to establish an association, or logical connection, between two TASE.2 instances. Such an association is typically long-running, staying in place as long as both TASE.2 instances are running and the underlying communications links are maintained.

Three operations are defined for Association objects:

1) Associate - used by a client to establish an association with a server.
2) Conclude - used by either a client or server to provide an orderly termination to an association (for example, for some planned maintenance or if there is a BLT or TASE.2 version mismatch).
3) Abort - used by either client or server to terminate an association when there are failures in the underlying communications mechanisms.

There are no actions defined for Association objects.

7.2 Data Value

Data Value objects represent values of control center data elements, including SCADA points, such as analog measurements and digital status, or data structures. Any data element or object that is uniquely identified by a single MMS Named Variable (with persistence) could conceivably be represented via the Data Value object. However, the only objects currently defined as Data Value objects are Indication Point, Protection Equipment Event objects, and Tag Value[1].

There are four operations defined for Data Value objects:

1) Get Data Value - can be used to request the value of a single SCADA point.
2) Set Data Value - intended to permit a data value to be written or set at a local control center by a remote control center. In practice, few vendors or utilities actually permit the Set capability in a TASE.2 client because of the desire to keep the ability to change data with the owner of the data, which will be the TASE.2 server. Note that the Device object defined below is intended to permit remote supervisory control operations.
3) Get Data Value Names - allows client to obtain a list of the names of all the Data Value objects at a remote control center for which that client has permission (via the BLT). This operation can be used to determine which points can be viewed by the client as an aide in defining data sets or one shot requests for data, as described later.
4) Get Data Value Type - allows client to obtain the Type attribute for a Data Value object.

There are no actions defined for Data Value objects.

[1] The second edition of TASE.2 added Tag Values to the objects considered as Data Value objects.

7.3 Data Set

Data Set objects are ordered lists of Data Value objects maintained by a TASE.2 server[2]. This object enables a client to remotely define Data Sets via TASE.2. The Data Set object can be used by a client, for example, to remotely define a list of SCADA points to be reported as a group. A specific Data Value object can be included in more than one Data Set if desired. The establishment of the reporting criteria and the actual transfer of data vales is accomplished using the Transfer Set object, as described below.

There are six operations defined for Data Set objects:

1) Create Data Set - allows a client to create a Data Set object at a remote server. In addition to specifying the list of Data Value objects to be included in the Data Set, the client can also specify which of the following parameters (referred to as Special Transfer Objects) will be included in a Transfer Report containing the actual data values:
   - Transfer Set name - identifies the Transfer Set object that generated the report.
   - Data Set Conditions Detected - identifies the event that triggered the sending of the report. The list of possible trigger events is:
     - Interval time-out;
     - Object change;
     - Operator request;
     - Integrity time-out;
     - Other external event.
   - Event Code Detected - identifies the event code if the trigger was Other External Event (see B. above).
   - Transfer Set Time Stamp - specifies the time the Transfer Report was generated at the server.
2) Delete Data Set - allows a client to delete a previously defined Data Set object.
3) Get Data Set Element Values - allows a client to obtain the value of each of the Data Value objects included in the referenced Data Set object. This operation permits a one-shot request of all values of the list of Data Value objects included in the referenced Data Set.
4) Set Data Set Element Values - allows a client to set the value of each of the Data Value objects included in a Data Set. In practice, this is not usually permitted.
5) Get Data Set Names - allows a client to get the names of all the Data Set objects currently defined at a server.
6) Get Data Set Element Names - allows a client to obtain the list of names of all the Data Value objects currently included in a specific Data Set object at a server.

There are no actions defined for the Data Set object.

Typical use: transfer of SCADA data to and from a real-time SCADA database on an EMS/SCADA system.

---

2 Since Tag Values were added to the objects considered as Data Value objects in the second edition of TASE.2, Tag Values can be reported in data sets whenever they change or periodically, rather than having to request current values via polling individually with the Get Tag operation defined for Device objects, which was the only mechanism provided in the first edition of TASE.2.

7.4 Transfer Set

Transfer Set objects residing at a TASE.2 server are used by a TASE.2 client to establish the actual transfer of data values. While Data Value objects can be individually requested via a one-shot request, receiving the requested value in response, more complex data transfers require the use of a Transfer Set. In particular, the transfer of groups of data defined in Data Set objects requires the use of a Transfer Set. The exchange of most all other data in TASE.2 requires a Transfer Set to be established first.

The Transfer Set object permits information to be exchanged on a periodic basis, on change of state or value, in response to a particular server event, or on operator request. The Transfer Set object provides the operations needed by a client to set up instances of Transfer Sets for each desired data exchange.

7.4.1 Four Transfer Set Object Models

Because of the unique requirements for transferring different types of data between control centers, TASE.2 provides four types of Transfer Set objects:

1) **Data Set Transfer Set** - used for establishing the transfer of Data Sets defined and created using the Data Set object.
2) **Time Series Transfer Set** - used for transferring the data values of a single Data Value object at different incremental times as specified by a delta time interval
3) **Transfer Account Transfer Set** - used for transferring many different types of data objects. In TASE.2, a Transfer Account is a generic term applied to a whole class of data objects used to represent information on schedules, accounts, device outages, curves, and other entities used by control centers which have only one thing in common – the use of complex data structures to represent data. Initially, the type of data envisioned was accounting or scheduling data, which represent an amount of energy transferred from one utility to another on a periodic basis, hence the name Account or Transfer Account.

   As currently defined in IEC 60870-6-503, this transfer set is used to transfer any of the data objects defined as “Block 8 objects”. This includes the following:

   — **Transfer Account** - this is a container type of object which can be used for the exchange of any periodic or profile data for control center energy scheduling, accounting, or monitoring applications.

   — **General Data Report** - this is another container object that can be used to exchange report data with control. 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, there is no temporal relationship between rows.

   — **General Data Response** - this object is used to send a response to a Block 8 General Data Report object or any other Block 8 object that needs an acknowledgement. It further supports transactions involving multiple data transfers.

   — **Device Outage** - this is a data object designed to exchange information about device outages, either for scheduling outages or reporting actual outages. Devices can include almost any type of physical component in a power system that is routinely monitored for status today.

   — **Availability Report** - this is the first of five data objects included in a class of data objects labelled Power Plant objects in IEC 60870-6-802. It is intended for a power plant control system or GCSs to report on predicted availability of generating units and/or to schedule outages. It is similar to the Device Outage object, but differs by having more attributes unique to generation units and power plants and by not including actual status reports (this is handled by the next data object, Real Time Status).

   — **Real Time Status** - this object is used by a power plant to report the actual operating status of generating units at the time of the report.

— **Forecast Schedule** - this object is intended for use by an EMS or GCS to deliver a forecasted usage of generating units at a power plant. Similar to the Transfer Account data object, this is another container object with user-defined matrix to specify the number and meaning of each column in the matrix. Rows are separated by a user-selected delta time increment.

— **Curve** - this object is intended for use by a power plant to report various types of curve data, such as heat rate, IO, incremental heat rate, MVAR capacity, opacity, SOX, NOX, and CO2 emission curves. The curve is represented as a sequence of curve segments, with each segment defined in terms of a polynomial.

— **Power System Dynamics** - this is a collection of data elements (rather than an actual object model) which need to be exchanged between a power plant and a GCS or EMS. These are scalar quantities and can be represented individually as Data Value objects.

These objects are described in detail in this guide in Subclause 8.8.

4) **Information Message Transfer Set** - used for transferring the Information Buffer data object defined in IEC 60870-6-802. The Information Buffer is intended for sending unstructured ASCII text strings or binary data. It could also be used for sending XML files or other encoded files.

There are four operations defined for Transfer Set objects:

1) Start Transfer - permits a client to request a server to begin to transfer data under the conditions specified by the client in this operation. The capabilities provided differ in important ways for each type of transfer set:

   – For Data Set Transfer Sets, the client provides the name of the Data Set object to use for grouping Data Values for transfer. A separate Transfer Set is used for each Data Set of interest, permitting different transfer conditions for each. A Data Set Transmission Parameters object is used to specify the transfer conditions, which include a start time, whether to report periodically or by exception only, an interval time between reports for periodic transfers, a buffer time for reporting by exception that starts when the first object change occurs, and whether the transfer is deemed “critical”, which causes an acknowledgement to be returned upon successful receipt of the transfer report. The second edition of TASE.2 also added the capability of requesting all changes in a specific data value object to be reported rather than just the last change. A Data Set Conditions Reported object permits the client to request that the conditions which triggered the transfer report also be included in the report.

   – For Time Series Transfer Sets, the client names the Data Value object of interest.

   – For Transfer Account Transfer Sets, the client can only enable the transfer all Transfer Account objects defined in the Bilateral Table. That is all Block 8 objects get enabled at one time and under one set of conditions.

   For Information Message Transfer Sets, similar to Transfer Account Transfer Sets, the client can only enable all Information Messages under the same set of conditions.

2) Stop Transfer - used by a client to stop a data transfer operation (i.e., disable the transfer). A new Start Transfer operation is required to once again enable the transfer.

3) Get Next Data Set Transfer Set Value - used by the client as the first step in starting a Data Set data transfer. The server maintains a “pool” of available Data Set Transfer Sets for a client to use. The client must obtain the name of the next available Transfer Set, and then perform a Start Transfer operation using the name of that Transfer Set to actually start a transfer. Thus the Start Transfer operation may be thought of as the client “writing” a value of the Transfer Set variable to the server. A Stop Transfer operation actually releases the Transfer Set back into the pool of available Transfer Set names at the server.

4) Get Next Time Series Transfer Set Value - similar to the Get Next Data Set Transfer Set Value operation, only this operation is used for starting the reporting of a series of values for the same Data Value object.

NOTE There are no “Get Next Transfer Set Value” operations for Transfer Accounts (i.e., Block 8 objects) or Information Message objects, since the client can only start or stop transfers of all Block 8 objects or Information Message objects, respectively.

There are two actions for Transfer Sets:

1) Condition Monitoring - performed by the server for each Transfer as soon as that Set that is Enabled via a Start Transfer operation. Any and all conditions requested in the Start Transfer operation are monitored by the server until a Stop Transfer operation is performed by the client. Note that for Information Message Transfer Set objects, the conditions used are locally defined only and cannot be specified via the Start Transfer operation.
2) Transfer Report - a Transfer Report is generated whenever a condition specified by the client has occurred for an enabled Transfer Set. The Transfer Report is the action used to actually transfer data from the server to the client. The server formats and sends a report with the appropriate data for the type of Transfer Set.

Included in the Data Set Transfer Report action may be any of the four Special Transfer Objects specified in the Data Set Create operation to convey information about the Transfer Report generation process. If the Transfer Set has the Report By Exception (RBE) attribute of DSTransmissionPars set to true, then these Special Transfer Objects may be included in the Transfer Report:

— Transfer Set Name - the name of the Transfer Set object which caused the Transfer Report;
— Transfer Set Conditions - a bitstring indicating which Transfer Condition(s) triggered the transfer;
— Transfer Set Time Stamp - the time of generation of the Transfer report;
— Transfer Set Event Code - indicates the external event which caused the Transfer Report to be sent, if the Other External Event condition was being monitored.

7.5 Account

Transfer Account objects (i.e., Block 8 data objects) are usually transferred via the Transfer Account Transfer Set object. However, the Account object provides one operation, the Query Operation, that permits a client to request a particular account object based on the account reference number and optionally start time and duration.

7.6 Device

Device objects represent actual physical devices in the field for the purpose of providing services for a client to control them remotely. Both interlocked (i.e., select-before-operate) and non-interlocked devices are represented.

There are four operations for Device objects:

1) Select - used by a client to request selection of an interlocked device only. If successful, the Device state is changed from IDLE to ARMED by the server.
2) Operate - used by a client to send a command to a Device object to execute a function. For interlocked devices, the Device state must be ARMED.
3) Set Tag - used by a client to set the Tag attribute of a Device object.

4) Get Tag Value - used by a client to retrieve the current state of the Tag attribute of a Device object[3].

There are four actions defined for Device objects:

1) Time-out - results from a time-out after a device has been set to ARMED via a Select operation but not yet operated. This action causes the device state to return to IDLE.
2) Local Reset - causes a device state to be reset from ARMED to IDLE by a local action at the server. This may also cause the Tag attribute value to change.
3) Success - used to tell the client that a successful Operate operation has been completed.
4) Failure - used to tell the client that an Operate operation has failed.

Direct communication of these actions to the client requires the use of the Event object, as explained in Subclause 7.8 below.

7.7 Program

A Program object provides a client with remote operation of a program at a server site. The actual program being controlled can be any application program at the server site.

In practice, the Program object is not used in utility-to-utility connections, since a more common pattern is for the server site to use some event it receives acting as a client (for example, change in data value or state) to trigger the execution of the appropriate program, thus maintaining control of program execution at the server site rather than from a remote client.

However, there may be cases where remote control of an un-manned site owned and operated by the same utility is, in fact, the capability needed. That is the intended use of this object.

There are six operations defined for the Program object:

1) Start - starts an IDLE program;
2) Stop - stops a RUNNING program;
3) Resume - starts a STOPPED program;
4) Reset - IDLEs a STOPPED program;
5) Kill - makes a program UNRUNNABLE;
6) Get Program Attributes - returns information on a RUNNING program.

There are no actions defined for a Program object.

7.8 Event

An Event object represents a system event at a server site, such as a device changing state or the occurrence of a certain data error. Event objects provide a way for a client to be notified of system events at a server. There are actually two objects associated with events: Event Enrollment object and Event Condition object. There is only a minimal description of these objects in the TASE.2 specification, which map directly to MMS services with the same name.

The types of events supported by the Event Condition object that a client can enroll to receive are defined in Subclause 8.5 of IEC 60870-6-503. They are concerned either with access violations, data failures, or events associated with operation of Select Before Operate (SBO) devices when using the Device object as described in Subclause 7.6 above.

---

[3] Note that the current state of the Tag attribute can also be obtained by including a Tag Value object in a Data Set.

7.8.1 Event Enrollment

Event Enrollment permits a client to express interest in being notified of a particular event when it occurs at a server site. There are three operations associated with an Event Enrollment object:

1) Create Event Enrollment - creates an Event Enrollment object which specifies which event is of interest and which conditions should be reported. This is accomplished by specifying the name of an Event Condition object as a part of creating an Event Enrollment object.
2) Delete Event Enrollment - deletes an Event Enrollment object.
3) Get Event Enrollment Attributes - gets existing Event Enrollment attributes.

There are no actions defined.

7.8.2 Event Condition

Event Condition objects are predefined at a server for all system events that are to be available to clients for enrollment.

There is one action for an Event Enrollment object:

1) Event Notification - notifies all clients that have created Event Enrollment objects that specify the particular Event Condition object whenever the event occurs.

   It should be noted that the device state change events that are monitored by Event Condition objects may also be reported to a client via SCADA data point changes, so that the use of Event objects may not be needed. However, the Event objects provide a mechanism for certain events that may not otherwise be reported to a client.

There are no operations defined for the Event Condition object.

8 Conformance Blocks and Associated Objects

This clause explains the intended use of each conformance block and object. The services and protocols associated with each conformance block and its associated objects are discussed in IEC 60870-6-503. The user objects themselves are described in IEC 60870-6-802. There are location references at the beginning of each block's description that point to discussions or descriptions in IEC 60870-6-503 and IEC 60870-6-802.

TASE.2 was from the beginning designed to be modular. Each conformance block represents a specific function or set of functions that a utility might wish to implement. A utility implementing TASE.2 for real-time data exchange is only required to purchase block 1. Additional blocks may be added independently. For example, a utility wishing to exchange power system data by exception and accounting data only needs to purchase blocks 1, 2 and 8.

Each block may have specific user objects associated with that block. This mapping of which objects are associated with which conformance blocks is found in IEC 60870-6-802, clause 9. When a user decides to purchase a specific block, they should also specify which objects within that block must be supported by the vendor.

8.1 Block 1 (Periodic Power System Data)

Block 1 is slightly different from all the other blocks. Block 1 is the minimum that a developer can implement. It is also the minimum that a user can purchase. There are certain system services that must be supported. In particular, this block includes the following objects:

— Association
— Data Value
— Data Set
— Data Set Transfer Set

Once these objects and associated services are provided in Block 1, they will be utilized whether additional Conformance Blocks are added or not.

With these system services, Block 1 provides for the periodic transfer of power system data. Power system data is the database representation of field device status (i.e. breakers, MODs, HLO lamps, substation doors, etc.), analog values (i.e. megawatt, megavar, voltage, tap settings, phase shifter angles, etc.), and accumulator values (KWH, KQH). Each data item may also have a quality code that provides information about the reliability of the data item itself.

The data object used in Block 1 to transfer this power system data is the Indication Point Object. Two optional data objects transferred in Block 1 are the following:

- Protection Event object.
- Tag Value object.

These objects are described in the following subclauses.

8.1.1 Indication Point Object

The Indication Point Object is used to transfer information about status points (represented as either STATE or DISCRETE data types), analog points (represented as REAL data types), and counter values (represented as DISCRETE data types).

A formal description of the Indication Point object can be found in IEC 60870-6-802, Clause 5.1.1.

8.1.1.1 Status Points

A description of the Status Points foundation types can be found in IEC 60870-6-802 subclause 6.1.1 as Data_State and Data_Discrete.

The user should decide whether to transfer status point information as STATE or DISCRETE. Using STATE will only allow up to a maximum of four states to be described for each device. Most power system devices are two or three state devices (open, closed, travelling). The choice of STATE allows for the most efficient transfer of status information. Two bits are used to encode the device state. The entire device state and quality are transferred in one octet.

There are, however, multi-state devices and pseudo status points in the SCADA/EMS database that have more than four states. To transfer these status points, the use of DISCRETE is required. Although less efficient, the use of DISCRETE allows the user to transfer a 32 bit integer where each value can represent a different state. Transferring status information using DISCRETE requires a 32 bit integer for the device states and an additional octet for the associated quality codes.

To promote interoperability, it is recommended that type State should be used to represent one and two bit status values and type Discrete for multiple status values of over two bits.

The Indication Point object when used for status is generic in nature in that more than one type of device can be represented with these object models. Clause 8 of IEC 60870-6-802 provides the allowable uses of these object models to represent real devices. That means that the values to be used for STATE (00, 01, 10, 11) and for DISCRETE (0,1,2,3) are normative

and the semantics are normative as far as they go for the devices listed. This means that if a device has two states, OFF and ON, then the standard requires for type STATE that these meanings be assigned to the values 01 and 10, respectively, and for DISCRETE to be assigned to 1 and 2, respectively.

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 implementors can add new semantics as long as they do not conflict with the existing semantics assigned to values in this subclause. For example, if a new device requires for example 5 states with values OFF, LOW, MEDIUM, HIGH, and OVER LIMIT, which cannot be mapped into the existing values and semantics, then it is permitted to add new values and semantics.

8.1.1.2 Analog Points

Analog inputs may be represented as either type Real or type Discrete. Type Discrete should be used to represent analog raw values which have not been scaled. Type Real should be used to represent analog values which have been converted to engineering units (i.e., when scaling and normalization procedures are done at the TASE.2 server end).

A description of the Real Analog Points foundation type can be found in IEC 60870-6-802 subclause 6.1.1 as Data_Real.

Analog point values are transferred as 32 bit IEEE format floating point values. Each analog value may have associated with it quality codes that provide information about the reliability of the value itself. Transferring analog information requires a 32 bit integer for the analog value and an additional octet for the associated quality codes.

8.1.1.3 Quality Codes

A qualitative description of the quality codes that TASE.2 provides to the user is found in IEC 60870-6-802 subclause 5.1.1 as Validity.

A description of the Quality Codes foundation type can be found in IEC 60870-6-802 subclause 6.1.1 as Data_Flags.

Quality codes are derived from the current SCADA/EMS computer system's ability to determine the reliability of a status, analog or accumulator point that has been stored in the SCADA/EMS database.

A telemetered value within reasonability limits that was updated to the SCADA/EMS database successfully on the last attempted scan has the highest quality. Its quality is derived from the fact that the value is both accurate and current. Quality is also considered high on data points that may not be current, but that have been manually entered by a dispatcher, operator or program. Because a "conscious" decision has been made to assign a point its particular value, it is considered "good" or of high quality.

TASE.2 transfers the quality codes associated with each data point; however, the mapping of local quality code bits in the receiver's SCADA/EMS database to these TASE.2 quality codes is a local implementation issue. Because each SCADA/EMS has its own symbols for displaying data quality, each user must determine their own hierarchy of processing and mapping to their own quality symbols.

8.1.1.4 Time Stamp

A description of the Time Stamp foundation types can be found in IEC 60870-6-802 subclause 6.1.1 as Data_TimeStamp and Data_TimeStampExtended.

The Timestamp attribute is used to assess the currency of the data value being transferred. Data can be "old" for a number of different reasons: delays in out going queues at the source SCADA/EMS, delays in transmission across the network, delays due to congestion and re-transmission within the network and delays in in-coming queues at the receiving SCADA/EMS. For all of these reasons, the data might need to be time stamped at the source SCADA/EMS at the earliest time following collection of that data from the field device. Values that are calculated from other values in the SCADA/EMS should be time stamped at the time the values are stored in the SCADA/EMS database. However, the actual time used in the time stamp is a local implementation issue.

The first edition of TASE.2 provided for just one Time Stamp type – Data_TimeStamp, which provides 1 s resolution, which is adequate for most SCADA data transfers. However, the second edition of TASE.2 adds a new Time Stamp type – Data_TimeStampExtended. The new type is optional and is to be used when there is a need for 1 ms resolution. This could occur, for example, when the time tag is to be used for sending Sequence Of Events (SOE) data, as for substation or power plant communications, or when TASE.2 is used to relay SOE data from one control center to another.

8.1.1.5 Change of Value (COV) Counter

A description of the Change of Value foundation type can be found in IEC 60870-6-802 subclause 6.1.1 as COV_Counter.

A periodic information report transferring status and analog values will normally transfer only the current value of the data point (i.e., the last value received before an information report if more than one value was received since the last information report)[4]. A receiving control center might want to know whether the point had changed and then changed back between information reports. For example, an auto-reclose operation might easily occur between information reports and not be recorded at the receiving site. A COV counter is incremented each time the owner sets a new value for the Indication Point.

8.1.1.6 Building Complex Data Types

The complex types are created by combining foundation data types. The choice of which complex data type to use is made by the implementer and is a balance between efficiency and the extent to which additional information about the value being transferred is required by the receiving site. For instance, if a client wants to receive status with quality codes and a time tag, the client would specify the use of the Data_StateQTimeTag complex type, described in IEC 60870-6-503, subclause 6.1.1.

8.1.2 Protection Equipment Event Object

The protection equipment event object definition can be found in IEC 60870-6-802 sub-clause 5.1.3.

When events occur at the substation, local relay actions may be taken to protect equipment. These events may be phase-to-phase, phase-to-ground, over current, over or under voltage, or other protective relaying schemes. The protection equipment event object model represents these 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.

---

4 The second edition of TASE.2 added an alternative to the use of the COV counter to convey that multiple changes in a data value object have occurred since the last information report. This new optional mode of data transfer permits the reporting all changes since the last information report, including a time stamp for each change if desired.

In addition to the name of the event, protection equipment event object reports:

1) The quality of the information. An underlined value indicates a 'yes' answer to the question.

ElaspsedTimeValidity	Were the associated times correctly acquired?	VALID INVALID
Blocked	Is the information blocked against further updates until it has been transmitted or safely saved?	NOTBLOCKED BLOCKED
Substituted	Was the information manually entered or entered by an automated source?	NONSUBSTITUTED SUBSTITUTED
Topical	Was the last update of the information successfully completed?	NONTOPICAL TOPICAL
EventValidity	Were no abnormal conditions of the information source detected during the last update?	VALID INVALID

2) The type of event (SINGLE or PACKED) and information related to the event.

A SINGLE event has its EventState, EventDuration and EventTime reported.

A PACKED event reports either the cause and involved equipment (START) or the actions taken (TRIP).

START events includes the following information: an underlined value indicates a yes answer to the question.

StartGeneral	Was this a general start?	START NOSTART
StartPhase1	Was phase 1 involved in the event?	START NOSTART
StartPhase2	Was phase 2 involved in the event?	START NOSTART
StartPhase3	Was phase 3 involved in the event?	START NOSTART
StartEarth	Was ground current involved in the event?	START NOSTART
StartReverse	Was reverse current involved in the event?	START NOSTART
DurationTime	Event duration in milliseconds	
StartTime	Protection equipment operation start time	

TRIP event includes the following information: an underlined value indicates a yes answer to the question.

TripGeneral	Was this a general trip operation?	TRIP NOTRIP
TripPhase1	Was a control operation issued to trip phase 1?	TRIP NOTRIP
TripPhase2	Was a control operation issued to trip phase 2?	TRIP NOTRIP
TripPhase3	Was a control operation issued to trip phase 3?	TRIP NOTRIP
OperatingTime	Time in milliseconds from the start of the operation until the first command was issued to an output control circuit	
TripTime	Time of the start of the operation	

8.1.3 Tag Value Object

The Tag Value object is described in IEC 60870-6-802, Subclauses 5.1.2, 6.1.2, and 7.1.2.

This object is derived from the Control Point object model used in Block 5 Device Control described in Subclause 8.5 of this technical report. Specifically, the Tag Value object is derived from the Tag and Reason attributes defined as part of the Control Point in IEC 60870-6-802 subclause 5.1.2. It indicates whether or not a Control Point 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, with the Reason attribute giving the reason for the tag. The Tag_Value structure is defined in IEC 60870-6-802 subclause 6.1.2.

8.2 Block 2 (Extended Data Set Condition Monitoring)

A description of Data Set condition monitoring can be found in IEC 60870-6-503 subclause 5.2.9.1.1 and 5.2.9.1.2.

Block 2 is used to provide the capability to transfer power system data in more ways than as just periodic reports. A periodic report (block 1) is simple and easy to set up, but it has the drawback that because it reports every value to the client every time the report is generated, it is not very bandwidth efficient. Block 2 is also referred to as report-by-exception, or RBE.

Report-by-exception allows the client to specify that power system objects will be reported only when a change is detected or when an integrity check is performed. TASE.2 does this by having the server monitor a number of conditions and when one or more of those conditions occurs, the data that has changed is sent to the client. The client sets the conditions to be monitored in the transfer set at the server.

The conditions that can be monitored are:

– The normal reporting period is due (IntervalTimeOut). This is the same condition that is monitored in block 1.
– The value, state or quality code of a value has changed (ObjectChange).
– The operator at the server site has requested that the value be sent to the client (OperatorRequest).
– A periodic report of all values is sent to the client to ensure that the two databases are still synchronized and that no changes have been lost since the last integrity check (IntegrityTimeOut).
– Other, unspecified conditions can be monitored (OtherExternalEvents).

Once the server has determined that a report by exception information report is required, it must then determine whether the client has requested that the report be generated as normal MMS named variables, or as blocked data (see next subclause).

8.3 Block 3 (Block Data Transfer)

A description of the rules for encoding block data can be found in IEC 60870-6-503 subclause 7.1.4.4.2.

Block data with report by exception can be an efficient transfer mechanism under certain conditions. It provides the possibility for a TASE.2 server to send power system data to a client with fewer bytes than required for sending with full ASN.1 encoding, as required in Blocks 1 and 2. Blocking may be useful where bandwidth is at a premium due either to low data rates or short periodicities (i.e., high frequency) of the data reports. However, the consequence of blocking is that the information needed to properly decode the data in a transfer report is not all contained in the report itself.

There are two mechanisms used by Block 3 to achieve efficiency. The first is the dropping of the tag and length fields for each data value reported. The second is the creation of an index-based tagging scheme to replace variable names with a one or two byte number. Block 3 provides three rules for encoding. The choice of the proper rule depends on whether the data is all sent periodically or as report-by-exception, and on how many values are sent. These mechanisms and rules are described below.

In practice, blocking is seldom if ever used, mostly since bandwidth is no longer a major consideration and long-term maintenance can be more of a problem than the more common non-blocking transfer techniques. Furthermore, under some conditions, given the efficiency of some types of ASN.1 encoding techniques, blocking can actually increase the number of bits on the wire.

8.3.1 Use of an Octet String MMS Variable

Instead of sending a tag and length along with each data value, as required by the ASN.1 Basic Encoding Rules used in Layer 6, the TASE.2 server instead utilizes a single long octet-string MMS variable to transfer all the data values. This requires that all primitive data types (and any aggregates based on them) be encoded using the full length permitted by that type in order to avoid putting in the length fields. Therefore, variable length fields must be padded out to their maximum length. In order for the client to receive and utilize the data, the client must have prior independent knowledge of the location and type of each value in the octet-string. Client knowledge of the type field is required to permit the dropping of the tag fields.

Then, if the data is sent as provided in Block 1 (i.e., not Report By Exception), the data is encoded into the octet-string according to rule 0, described below:

Rule 0: [rule#, total length, $value_i$ ...]

This can result in fewer bytes being transferred because the tag and length fields for each variable are not transferred. This works best for variables with short data types that require only one byte for the value. However, for longer types, there are cases where this will not result in any savings. For example, transferring an Integer32 variable that happens to equal 0 in value will result in a MMS PDU encoding using BER of 3 bytes (tag, length, and value each one byte) whereas blocking would have to expand the integer out to four bytes, actually wasting one byte. Therefore, the type of data to be transferred should be considered before automatically assuming fewer bytes will result just from dropping the tag and length fields.

8.3.2 Index-Based Tagging

Block data and report-by-exception can be combined to yield a more efficient transfer of data. If block data and report by exception are specified, the server has two rules available for constructing the message that will be sent to the client. In each case the database point is identified by an index into the named variable list, followed by the current value of the point. This has the effect of replacing variable names, typically many bytes in length, with a one or two byte index number.

Utilizing rule 1, the header consists of the rule 1, followed by the total message length in octets. The body of the message consists of a one octet index (the relative position of the identifier in the named variable list), followed by the value of the identifier. This pairing of index and value is continued to the end of the message.

Rule 1: [rule#, total length, $index_i$(1-octet), $value_i$ ...]

Rule 2 is similar to rule 1 except that it utilizes a two octet index for messages that have more that 255 index-value pairs.

Rule 2: [rule#, total length, $index_i$(2-octets), $value_i$ ...]

Blocking when combined with report-by-exception thus provides guaranteed efficiency for transmission by sacrificing inclusion of the information needed to decode the data contained in a message, creating a data maintenance task. If message formats seldom change, this may be a good tradeoff. However, if bandwidth is not a primary concern or report-by-exception is not used, and more flexibility is desired by a client to change the content of messages using TASE.2 protocol mechanisms without operator involvement at the server, then blocking should probably not be used.

8.4 Block 4 (Information Messages)

Block 4 provides a general message transfer mechanism that also includes the ability (by agreement of the two parties) to transfer ASCII text or binary files. Block 4 adds the Information Message Transfer Set server object with the associated Information Buffer data object.

One use of this service might be for a utility to notify other utilities within its inter-connection that an event more complex than that represented by simple power system data values, has occurred. For example:

– notification of a decision to implement an inter-connection wide time error correction action.

– notification of the boundaries of identified electrical islands during a disturbance.

– request for emergency use of pool reserves.

These messages might be simple formatted ASCII text messages with data from the SCADA/EMS incorporated into the body of the message. These could be used as alarm text or text reports for display on a receiving operator console or for logging.

The InformationBuffer object provides a unique identifier (InfoReference) and a local identifier (LocalReference). The MessageID identifies the particular instance of a message. The Size attribute is the length in octets of the actual data being transferred.

This object also provides a mechanism for simple, small, binary file transfer. These transfers are limited in size by MMS to the maximum PDU size negotiated when an association is established. The InfoReference and LocalReference attributes could be used to identify a process that would receive the binary information buffer and store it in a local file. The information stored could, by agreement, be a spreadsheet file or word-processing document that would later be accessed by the client or server. Individual instances of this file being transferred (for example, the June, July or August instances) would be distinguished by the MessageID attribute.

Another possible use would be to send files formatted as Extensible Markup Language (XML) files, which are ASCII text with the tag structure defined in the XML standard.

An informal description of the Information Message can be found in IEC 60870-6-503 subclause 5.1.6 and a formal description can be found in subclause 5.2.8. The Information Buffer object is described in IEC 60870-6-802 subclause 5.4, the type descriptions in subclause 6.4 and the mappings to MMS in subclause 7.4.

8.5 Block 5 (Device Control)

Block 5 adds the Device server object and associated Control Point data object.

Block 5 provides a mechanism for transferring a 'request to operate a device' from one TASE.2 implementation to another. TASE.2 does not directly control the device, rather it communicates a client's request to operate a device to the server.

TASE.2 retains some of the important characteristics of RTU device control. Specifically, the select and validate before operation for interlocked devices and the armed-for-execution mode for a selected device.

The ControlPoint object is used to transfer the request. It distinguishes between a device operation (COMMAND) and the transfer of a numeric value (SETPOINT), either floating point (SetpointRealValue) or integer (SetpointDiscreteValue).

A control request can be for non-interlocked (NONINTERLOCKED) or interlocked devices (INTERLOCKED). Both command and setpoint operations can be inter-locked or non-interlocked. Non-interlocked controls are control operations that do not require select-before-operate confirmation. These might include transformer tap changes, raise/lower operations and digital value setpoint type operations. Interlocked controls on the other hand, require select-before-operate confirmation for critical operations such as breaker trip/close, recloser on/off and HLO lamp on/off.

For interlocked control operations, the client sends a request to operate a specified device to the server. After checking for the existence of the device object, checking access control in the Bilateral Table, the server then performs a local verification that the device is available for operation. The verification checks that the server actually performs are a local implementation issue. The device is SELECTED and a previously agreed to CheckBackName is provided to the client to confirm that the correct device has been selected. A Time-out period is reported to the client giving the length of time that the device will remain selected by the server.

TASE.2 does provide a mechanism for the server to report to the client whether the desired control point is tagged. The server reports:

0 if the device is not tagged

1 if the device is tagged open and close inhibit

2 if the device is tagged close only inhibit

The client, having received a verification of device operability and a validation of device selected, then sends a final request to have the device operated by the server or to cancel the requested operation.

The server completes the requested control operation and notifies the client if the command was sucessfully received by the end device. The only ways to determine if the operation was successfully completed or not is (1) to scan an indication point associated with the control point in the EMS power system model (local implementation issue) or (2) you need to enroll to receive the SBO actions Success and Failure with the Event object in Block 7. In practice, the use of an associated indication point is usually used to provide the most reliable indication of success or failure. Provisions are made so that at any time during this process the client or the server can terminate the operation for a valid reason.

The ControlPoint object is generic in nature in that more than one type of device can be controlled with this object model. Clause 8.2 of IEC 60870-6-802 provides the allowable uses of this object model to represent specific device switching commands. That means that the values to be used for COMMAND (0, 1) are normative and the semantics are normative as far as they go for the devices listed. For example, the values assigned to TRIP and CLOSE for a switch are 0 and 1, respectively.

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 implementors can add new semantics as long as they do not conflict with the existing semantics assigned to values in this subclause.

An informal description of device control can be found in IEC 60870-6-503 subclause 5.1.10 and a formal description can be found in subclause 5.2.11. The device object model mapping can be found in subclause 6.15 of IEC 60870-6-503. Device operations and action mapping to MMS including a sequence of device control diagram can be found in subclause 7.1.6.1 of IEC 60870-6-503. The control point object is described in IEC 60870-6-802 subclause 5.1.2, the type descriptions in subclause 6.1.2 and the mappings to MMS in subclause 7.1.2.

8.6 Block 6 (Program Control)

Block 6 adds the Program server object and associated services.

Block 6 provides a mechanism for a TASE.2 client to perform program control at a server TASE.2 implementation site. Program control is only available by prior agreement between any two TASE.2 sites.

Implementation of program control is made very straight forward by the fact that MMS provides program invocation and control as part of its basic services. TASE.2 can then utilize these services with proper interfaces to the SCADA/EMS system to perform remote program control. There are no user objects associated with program control.

An informal description of program control can be found in IEC 60870-6-503 subclause 5.1.11 and a formal description can be found in Subclause 5.2.12. The program control object model mapping can be found in Subclause 6.16 of IEC 60870-6-503. Program operations and action mapping to MMS can be found in Subclause 7.1.7 of IEC 60870-6-503.

8.7 Block 7 (Event Reporting)

Block 7 adds the Event Enrollment and Event Condition objects. Block 7 is not required for any of the other blocks, but instead provides extended reporting of standardized system events occurring at a remote site (i.e., TASE.2 server) as defined in Subclause 8.5 of IEC 60870-6-503.

Block 7 provides two functions to the TASE.2 client.

1) It allows the TASE.2 client to enroll in three types of events:
   - a) Notification of server actions resulting from Device object SBO actions:
     - i) Time-out;
     - ii) Failure;
     - iii) Local reset;
     - iv) Success.
   - b) Access violations.
   - c) Data transfer failures.
2) It allows the TASE.2 client to receive information about the events that the client has enrolled in and has enabled.

An informal description of event reporting and event conditions can be found in IEC 60870-6-503 subclause 5.1.12 and 5.1.13, and a formal description can be found in Subclause 5.2.13 and 5.2.14. The event enrollment and event conditions object model mapping can be found in Subclause 6.17, and 6.18 of IEC 60870-6-503. Event enrollment operations mapping and event conditions action mapping to MMS can be found in Subclause 7.1.8 and 7.1.9 of IEC 60870-6-503.

8.8 Block 8 (Additional User Objects)

Block 8 adds the Transfer Account Transfer Set server object for transferring Block 8 data objects. An informal description of Transfer Account Transfer Set objects and services can be found in IEC 60870-6-503 subclause 5.1.7 and a formal description can be found in Subclause 5.2.9.3. The Transfer Account Transfer Set object model mapping is found in Subclause 6.9.3. Transfer Account Transfer Set operations and action mapping to MMS can be found in Subclause 7.1.4.

Block 8 also adds the Account server object for requesting Block 8 data objects. Using the Query operation supported by this server object, a TASE.2 client can specify the following:

– The Transfer Account Reference number for the account for which information is to be returned.
– Start time for the data.
– Duration of the data in seconds since the start time.
– A RequestID, which is echoed back by the server to permit the client to match incoming data with a specific request.
– TAConditions to identify the type of data requested.

A positive acknowledgement of receipt of a Block 8 object can be obtained through use of the new General Data Response object added in the second edition of TASE.2 (see Subclause 8.8.5 of this Technical Report).

An informal description of account objects and services can be found in IEC 60870-6-503 subclause 5.1.5 and a formal description can be found in Subclause 5.2.7. The account object model mapping is found in Subclause 6.7. Account operations and action mapping to MMS can be found in Subclause 7.1.5. However, the details of the attributes included in the Query operation are contained in IEC 60870-6-802 subclause 5.2.4.

Block 8 provides a utility with a number of additional *data objects* related to transferring scheduling and accounting information, device outage information, and power plant information. A vendor might offer support for one or more of the data objects in Block 8. Information on these objects can be found in the TASE.2 specification at the following locations in IEC 60870-6-802:

	Object Model	MMS Types	Object mapping to MMS
Scheduling and Accounting			
Transfer accounts	5.2.1	6.2.1	7.2.1
Transmission segment	5.2.2	6.2.2	7.2.2
Profile value	5.2.3	6.2.3	7.2.3
Account request	5.2.4	6.2.4	7.2.4
Device outage	5.3	6.3	7.3
Power plant			
Availability	5.5.1	6.5.1	7.5.1
Real-time status	5.5.2	6.5.2	7.5.2
Forecast schedule	5.5.3	6.5.3	7.5.3
Curve mapping	5.5.4	6.5.4	7.5.4
General Data Report[1]	5.6	6.8	7.6
General Data Response[1]	5.7	6.9	7.7

[1] This data object was added in the second edition of TASE.2.

The following subclauses describe the TASE.2 data objects in more detail.

8.8.1 Transfer Account Data Object

The ability to transfer scheduling and accounting information between TASE.2 implementations is a key feature of TASE.2. With it, a TASE.2 client can set up a transfer set that will allow the server to send pre-schedules, next hour schedules, mid-hour changes, after-the-hour-actuals and historical information. TASE.2 generalizes this transfer capability to allow any data that is collected on an hourly (or other period) basis, including such data as generator schedules, interchange schedules, forebay and afterbay elevations, average hourly TOT limits and actual flows, pricing information, delivery point loads, etc.

Profiles of a quantity or value that changes over time can also be sent with this object. The value may be energy, capacity, price and/or other information. It is represented by a series of ProfileValue objects, each of which specifies a target value, ramp start time and ramp rate. The transfer capability for both periodic and profile reporting is accomplished via the Transfer Account data object.

The flexibility of this object for periodic reporting is achieved through the use of a matrix data type with the number of rows and columns defined by the user for each type of desired transfer. In addition the meaning of the column headings are also user-defined, so this standard data object can be used to transfer many different types of schedules and accounts.

An important feature of TASE.2 is the ability of the client scheduler or dispatcher to specify the time frame for the data to be retrieved and have the server return the specified information corresponding to that time frame. The client specifies this via the TAConditions object described below.

The transfer account object definitions can be found in IEC 60870-6-802 subclause 5.2. A transfer account example is provided in Annex A of IEC 60870-6-802.

8.8.1.1 The Meaning of TA Conditions and How to Use Them

TAConditions (Transfer Accounts Conditions) were intended to allow the client to set up condition monitoring of specific accounts in the server. Within the electric power utility business, the transfer of scheduling and accounting information is very time dependent. To illustrate this time dependency, a utility might have the following scheduling and accounting time frame requirements:

– Pre-schedules must be completed by 16:00 the day before they are used.
– Next hour schedules must be completed by 20 minutes prior to the hour in which they will be used.
– Mid-hour changes can occur anytime within the current, already scheduled, hour.
– After-the-hour-actuals are due by 10 minutes past the just completed, previous hour.
– All 24 hours of historical forebay elevations for the previous day need to be transferred at 10 minutes past midnight of the current day.

In practice the TAConditions are used by the client to request a certain type of transfer account corresponding to the time frames described. The actual time frames used, however, are a local implementation issue. Since the exact formatting of the Transfer Account data object is user defined, it is possible to have a different format for data corresponding to each TACondition. The Transfer Account Reference number can be used to further identify the format of the report being transferred.

8.8.1.2 Transfer Account Structure

Scheduling and accounting data is stored by utilities in what is essentially a matrix structure. TASE.2 carries forward the matrix concept, but generalizes it to allow the user to define the meaning of the columns. Both floating point and integer matrixes can be transferred. This allows the TASE.2 client and server to exchange virtually any type of matrix format data in addition to scheduling and accounting data.

Generally speaking, the actual Transfer Report used to transfer a Transfer Account Data Object identifies the account to be transferred, the transfer account condition (TACondition), the sending and receiving utilities, the start time (referenced by hour ending, if hourly scheduling and accounting information is being transferred), and the time span of the period used (typically one hour). The message then specifies whether or not this is a wheeling transaction, and if so, the number of wheeling segments is identified. For each segment, the number of floating (or integer) values and the number of periods (the values that will form the matrix of information) are identified.

In the event that a matrix of information that has specific meaning to the client is being transferred, a list of local references can be identified which will provide the client system with the column heading information for the matrix.

8.8.1.3 Examples

Figure 5 illustrates the structure of this object. Up to one floating point matrix and one integer matrix can be sent in one Transfer Account object. Figure 6 shows the use of this object to send two matrices of data for a time period covering 1 h with a period resolution of 15 min.

A more detailed example of the use of the Transfer Account object is provided in Annex A of IEC 60870-6-802.

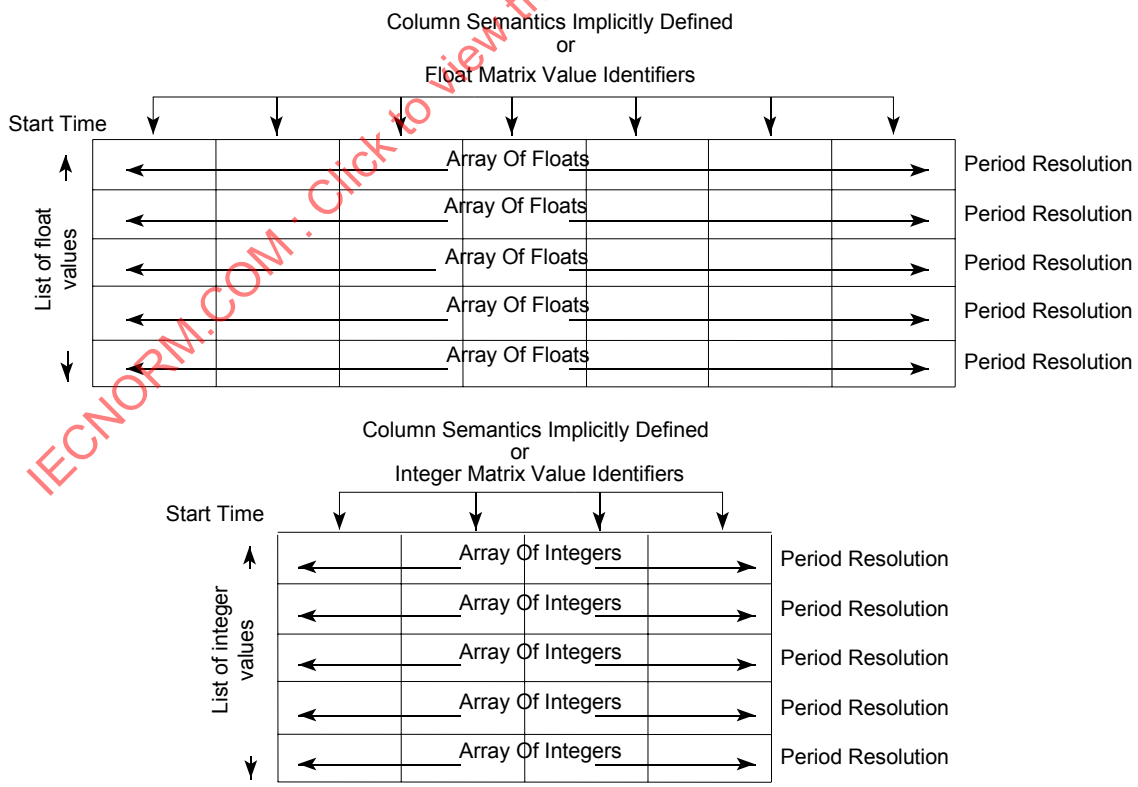

Figure 5 – Transfer Account Data Object Model Structure

StartTime: 10:00 AM
Period Resolution: 00:00:00:15:00
Column Semantics: Implied by TransferAccountReference

Floating Point Matrix

	Seller Cost	Buyer Cost	Emergency Cost	Tariff Value	Tariff Incurred	Tariff Avoided	Savings
10:00 AM							
10:15 AM							
10:30 AM							
10:45 AM							

Integer Matrix

	MWH	Emergency MWH	Area Load
10:00 AM			
10:15 AM			
10:30 AM			
10:45 AM			

Figure 6 – Example of Transfer Account Data Object Use

8.8.2 Device Outage

Utilities deal with two types of equipment outages, planned and unplanned. Planned outages of generators, transmission lines and line devices for maintenance and unplanned outages due to system disturbances which result in no, partial or full curtailments of power transfers. TASE.2 provides an object for the transfer of outage information.

The DeviceOutage object identifies the location of the device (StationName), the device (DeviceType) and provides information about the device (DeviceName, DeviceType and DeviceRating). If the outage is planned, then the object information can be used to set up a new outage or revise an existing planned outage. In either case, the out of service and return to service date and times are provided. The outage type can be categorized and if it is an outage resulting in a partial curtailment, new upper and lower operating limits can be specified.

If the outage is an Actual outage as a result of unplanned operation of devices in the electrical system that resulted in a loss of load, the type of action is categorized and the amount of load that was being carried at the time of the service interruption is provided.

The device outage object definition can be found in Subclause 5.3 of IEC 60870-6-802.

8.8.3 Power Plant Objects

TASE.2 has been extended to include communications between the SCADA/EMS and power plants. Power plants have specific requirements that result in specialized objects. These objects take advantage of the existing underlying TASE.2 services available such as report-by-exception and condition monitoring.

8.8.3.1 Power Plant Availability Report Object

The power plant availability report object is used to allow a generation station to inform the control center of the known or scheduled availability of a unit at that site. The object can also be used to schedule a unit outage or curtailment. If it is reporting a unit curtailment, new operating constraints can be reported for the period of the curtailment.

The object identifies the generation station location and the specific unit referenced in the report. It also identifies the start and stop time as well as the duration of the proposed change of status of the unit, either AVAILABLE or UNAVAILABLE. A curtailment is treated as AVAILABLE with a change of operating constraints.

If the unit is AVAILABLE, the report allows the generation station to specify for the duration of the availability:

- A new price.
- New maximum ramp rates both up and down.
- New gross maximum and minimum capacities.
- New net maximum and minimum capacities.
- Whether the unit is in standby or on-line mode:
  - if the unit is on-line, whether it is available for load following,
  - if the unit is on-line, but not available for load following, the reason (STARTUP, UNSTABLE).
- If the unit is UNAVAILABLE, the reason for its unavailability (FORCED, SCHEDULED, TESTING).

The reporting generation station can also report whether the unit is providing reserves, and up to 256 characters of user comments.

The power plant availability report object definition can be found in IEC 60870-6-802 subclause 5.5.1.

8.8.3.2 Power Plant Real Time Status Object

The power plant real time status object is used to allow a generation station to inform the control center of the current status of the plant and each unit at that site.

If the plant is AVAILABLE, the report allows the generation station to report the current operating characteristics of each unit.

- The maximum ramp rates both up and down.
- The gross maximum and minimum capacities.
- The net maximum and minimum capacities.
- Whether the unit is in standby or on-line mode:
  - If the unit is on-line, whether it is available for load following,
  - If the unit is on-line, but not available for load following, the reason (STARTUP, UNSTABLE).
- Whether the unit is externally blocked high or not.
- Whether the unit is externally blocked low or not.
- If the unit is UNAVAILABLE, the reason for its unavailability (FORCED, SCHEDULED, TESTING,EQUIPMENT).

The power plant real time status object definition can be found in IEC 60870-6-802 sub-clause 5.5.2.

8.8.3.3 Power Plant Forecast Schedule Object

Some power plants and units are operated with pre-scheduled base points. These units are either not used for load following or have pre-defined load following periods. This object allows base points and operating modes to be transferred to the generation stations. Modes of operation can be user defined in the event that load following or not load following does not adequately describe the required modes.

The scheduled period for which a megawatt base point value applies can be specified in the object. Normally, one hour (hour ending) would be used, but the user can define other scheduled period durations. The only constraint is that all the referenced periods have to be of equal duration.

The power plant forecast schedule object definition can be found in IEC 60870-6-802 subclause 5.5.3.

8.8.3.4 Power Plant Curve Object

The power plant may wish to transfer data in the form of two dimensional curves. These curves could be incremental heat rate, hydro head dependent efficiency curves, cost curves or other power plant related curves.

The curve is defined as an n^{th} order polynomial. All the segments in the curve are of the same order. Each segment of the curve is represented by a start of segment and an end of segment, and mathematically by the n^{th} order polynomial: $A_0 + A_1x + A_2x^2 \ldots A_nx^n$. In representing each segment's polynomial, only the coefficients, $A_0 + A_1 \ldots A_n$ are transferred. Knowing the order of the polynomials and these coefficients, the client can reconstruct the curve.

The power plant curve object definition can be found in IEC 60870-6-802 subclause 5.5.4.

8.8.4 General Data Report Object

After the publication of the first edition of TASE.2, it became apparent that there was a need for a more general container object in Block 8 that could accommodate floating point, integer, and text data values, without any implied relationship of time between rows, as is required in the Transfer Account object.

The flexibility of this object for reporting is achieved through the use of a matrix data type with the number of rows and columns defined by the user for each type of desired transfer. In addition, the meaning of the column headings are also user-defined, so this standard data object can be used to transfer many different types of report.

The General Data Report object is designed for sending one or more matrices of data. The object permits the transfer of up to two different integer, floating point, and text matrices in one object. The two matrices of each type can differ in the number of columns of values. This permits reports with one level of nesting for each type of matrix. The formatting of the report, based on the contents of the matrices, is defined by the recipient and can be different for each recipient, based on local preferences.

The General Data Report object also supports the concept of transactions which may span multiple exchanges through the use of a new attribute named Transaction Code. This code provides status on the current report as follows:

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

The General Data Report object definitions can be found in IEC 60870-6-802 subclause 5.6.

8.8.4.1 Example of General Data Report

An independent system operator may wish to send an AGC Regulation Capacity Report to one of the member companies. This is a simple example requiring only 1 integer matrix (see Figure 7).

- Integer matrix, 5 columns
- MatrixID1=UnitID
- MatrixID2=Regulation Type (A=1,B=2)
- MatrixID3=Assigned Regulation Class (I=1, II=2, III=3, IV=4)
- MatrixID4=Assigned Amount (MW)
- MatrixID5=Additional Amount (MW)

Integer Matrix

UnitID	Regulation Type	Assigned Regulation Class	Assigned Amount (MW)	Additional Amount (MW)

Figure 7 – Example of AGC Regulation Capacity Report using the General Data Report Object

8.8.5 General Data Response Object

In the first edition of TASE.2, Block 8 did not have a capacity to acknowledge receipt of a Block 8 object, such as the Transfer Account object. A related issue that arose was the need for conducting transactions with multiple states covering the transfer of several messages. This can arise with the transfer of Block 8 schedules (for example, energy or outage schedules) where, for example, a control area may submit a schedule to a power pool, the power pool needs to acknowledge receipt, then later indicate acceptance/rejection, and then the control area may need to submit a revised schedule, subject to the same approval/ rejection cycle again.

The second edition of TASE.2 added a General Data Response object to provide an acknowledgement for Block 8 objects. This object also provides an attribute named Response Code for indicating the transaction state of the Block 8 object. The Response Code is the numeric response code associated with the response. It has values which correspond to permitted responses to the Transaction Code defined in the General Data Report 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

The General Data Response object definitions can be found in IEC 60870-6-802 sub-clause 5.7.

8.9 Block 9 (Time Series Data)

Block 9 adds the Time Series Transfer Set server object.

Block 9 provides a TASE.2 client with the ability to receive time series data. Time series data might be data that has a required sampling time too fast to conveniently transfer it continuously between TASE.2 implementations and is not needed at the client site in real time. Examples of this type of data might be a 200 ms sample on key analog values on the backbone transmission system during a disturbance. The values once collected might then be transferred as historical data to a disturbance analysis center. Real-time trending of values with longer reporting periods for more efficient use of communications bandwidth is another potential time series data application.

In setting up a time series data transfer, the client establishes the begin time and the end time. Both begin time and end time may be in the past, in which case the server immediately generates a report of historical values based on the time frame specified. If the begin time is the current time or 0, the server begins to immediately collect values until the end time occurs. At the designated end time, the server stops collecting values and generates a report for the client. If the end time is the current time or 0, the server assumes current time and stops collecting values and generates the report. For a planned collection of values in the future, the client can specify both future begin and end time.

Two intervals are specified to make the collect and transfer of information as efficient as possible. The sample interval specifies the collection rate at the server. The reporting interval specifies the time between reports.

Several options exist to fine tune the reporting functions of the server. Since the end time may not fall exactly on one of the reporting period boundaries, the server can be instructed to generate a report at the specified end time. Similarly, the server can be instructed to generate a report at the initial enabling of the transfer set and at each reporting period. The server can also be instructed to allow an operator at the server system to initiate on demand a transfer of the time series data collected since the last reporting period.

The time series transfer set object model can be found in IEC 60870-6-503 in subclause 5.2.9.2 and the time series transfer set object model mapping can be found in Subclause 6.9.2.

9 Mapping Utility Data to Conformance Blocks and Control Center Data Objects

A utility planning to use TASE.2 must perform several tasks before knowing exactly which Conformance Blocks are required to meet its needs. The following issues will need to be addressed:

1) The data to be transferred needs to be identified and performance requirements established. This will uncover which TASE.2 services are needed, and hence which TASE.2 server objects and Conformance Blocks are needed. For instance, if SCADA data, schedule/accounting data, and text reports are to be transferred, then Conformance Blocks 1, 2, 4, and 8 will be required.
2) Analyzing data to be transferred will also identify which data objects are needed. For instance, if in addition to scheduling/accounting data, outage information for scheduling and reporting outages is needed, then either the Device Outage object or a combination of the Availability and Real-Time Status objects will probably be required.
3) Each data element will need to be mapped to a TASE.2 object attribute. This mapping will need to be documented. Every attempt should be made to map to existing standard objects specified in the TASE.2 specifications, since this will ensure interoperability with other TASE.2 vendor products without additional software development.
4) Some types of data reports will not map 1:1 to standard TASE.2 objects. The choice will then be one of the following:
   a) "Force fit" a data element to an attribute with a different meaning. This eliminates any TASE.2 software changes, but creates the opportunity for misunderstandings of meanings of attributes.
   b) Add a new attribute to a standard object, thus customizing it. If no new data types are introduced, this will result in minimal change, but will ensure there is no misunderstanding. Eventually, common usage may result in a modification of the standard object.
   c) Create a whole new data object. Sometimes, this is the only choice. This process is described in the next subclause.
5) All the choices made will need to be documented. A Network Interface Control Document (NICD) is commonly used for this purpose. This document is not defined anywhere since it goes beyond what is specified in the TASE.2 specifications, but by common practice it includes the mappings, the common conventions decided on for assigning numbers or codes to reference numbers, the definition of any new data objects, etc. For data objects that use the Matrix data type, the meanings of column headings and the number of rows for different uses of this object will also be documented here.

10 Definition of New Data Objects

The designers of TASE.2 expected that new objects would be needed from time to time. All possible uses of TASE.2 could not be envisioned during the creation of the initial revision of the specification. If the process of mapping actual data requirements to TASE.2 objects described in the previous clause requires a new object, then the process described in this clause should be followed.

The process for creating a new data object is as follows:

a) Develop an abstract data model, creating a name for the object and deciding which attributes are needed along with a definition of each attribute. The object model definitions in the TASE.2 specification provide examples of how this is done. This can be done by end users with little or no knowledge of MMS.

b) The abstract model then needs to be mapped to concrete structures with components. Part of this process is assigning data types to each attribute. The goal is to reuse the data types already defined in the TASE.2 specification, thus minimizing the implementation effort for the new object.

11 Using the PICS

Subclause 9.2 of IEC 60870-6-503 contains a Protocol Interface Conformance Specification (PICS) table which lists all the conformance requirements (i.e., model definitions and attributes, operations, and actions) that have been defined for TASE.2. The requirements fall into two categories: Mandatory and Optional. The requirements are grouped according to the TASE.2 server objects. A vendor must implement all Mandatory features for an object and may implement the Optional features.

An end user needs to specify which Optional features are desired when procuring or implementing a TASE.2 instance. Typically this would be specified in detail in a software Functional Specification for TASE.2.

12 Bilateral Table Issues

TASE.2 specifies access control through the use of Bilateral Tables. The functionality required is clearly presented in IEC 60870-6-503. The type of access for each TASE.2 data object is defined via these tables. However, implementation is left as a “local implementation issue.” This includes the management and maintenance of these tables. As a result, each vendor is free to choose how to implement the functionality for Bilateral Tables, including what type of operator interface to provide. This means that an actual physical table is not necessarily required, as long as the functionality is implemented according to the TASE.2 specification.

A common request from end users is to have a capability for a TASE.2 client to view the Bilateral Table at a TASE.2 server to determine which objects it has access to and to be notified whenever that table is updated. The TASE.2 specifications do not specify a way to accomplish this, although there is a Data Value operation, Get Data Value Names, which a TASE.2 client can use to obtain a list of all the Data Value objects accessible to that client. A “browser” capability could be quite useful for a user acting as a TASE.2 client, which would let a client view the objects it has access to, and then simply point and click which objects it wants to receive. Part of the functionality could include creating data sets by pointing and clicking. This would minimize the potential for operator error on entering data values. In any case, either of these approaches could result in a large amount of data being transferred, which could temporarily overwhelm a network not sized to take this into account. This could result in possible association failures for other associations sharing the bandwidth. Therefore, careful consideration should be given when planning for such capabilities.

Another approach is to establish a form for a utility to make the data values it is willing to share available and then post or email to other interested utilities, updating the contents as changes occur over time.

Some users may not desire to use the security mechanisms provided through Bilateral Tables, for instance, where TASE.2 is used between two regional control centers within the same utility. One way to handle this is just to provide the same access to all control center objects in the VCC to any client. However, the protocol operations and actions specified in the TASE.2 specifications must still be implemented to ensure interoperability.

13 User Interface Issues

The TASE.2 specifications do not specify a user interface for managing and maintaining TASE.2. This is left as another “local implementation issue.” Each vendor is free to choose an appropriate interface.

The following areas may need a user interface:

– Displaying TASE.2 performance data, such as status of each association and data link, last error detected, throughput statistics, etc.

– Control of data link associations, data sets, or other TASE.2 objects to enable or disable selected capabilities.

– Creation and editing of Bilateral Tables.

– Creation and editing of Data Sets.

– Setting up and managing broadcast groups for information messages when Conformance Block 4 is implemented. Since TASE.2 does not provide a broadcast capability, it may be desirable to have the ability to create broadcast groups that would specify groups of destinations (i.e., other TASE.2 sites or operator consoles) to receive information messages.

14 Other Local Implementation Issues

TASE.2 is a standard real-time data exchange protocol. It provides numerous features for the delivery of data, monitoring of values, program control and device control. All the protocol specifics needed to ensure interoperability between different vendor’s TASE.2 products have been included in the specifications.

The TASE.2 specifications, however, do not attempt to specify other areas that will need to be implemented in a TASE.2 software product but that do not affect interoperability. These areas are referred to as “local implementation issues” in the specification. TASE.2 implementers have the freedom to handle these in different ways and can therefore differentiate their products by the way they handle these issues. For example, one vendor may have a graphic-oriented user interface permitting point and click operations for creating data sets or controlling TASE.2 data links, while another may provide only programmer’s editing tools to accomplish these tasks.

Local implementation issues in the specification include but are not limited to the following:

– The API through which local applications interface to TASE.2 to send or receive data.

– A user interface to TASE.2 for user management of TASE.2 data links.

– Management functions for controlling and monitoring TASE.2 data links.

– Failover schemes where redundant TASE.2 servers are required to meet stringent availability requirements, such as those typically experienced in an EMS/SCADA system environment.

– How data, programs or devices will be controlled or managed in the local SCADA/EMS to respond to requests received via a TASE.2 data link.

These responsibilities fall to the SCADA/EMS vendor and the implementing utility. This clause will attempt to address some of the areas that have been specially identified as local implementation issues in the TASE.2 specifications that have not been covered elsewhere in this guide.

14.1 Client Server Association Management

The client always initiates the association establishment procedure with a TASE.2 server. A single TASE.2 site can act as both client and server to one or more TASE.2 sites. It can also simultaneously be just a client or just a server to other sites. In the case where it is both client and server with another site, the use of the associations between the two sites is a local implementation issue. One method is where each client uses a different association with its server. The other method is to utilize the same association for the client-server pairs in both directions. The ICCP specification requires support of single use association. The support of dual use associations is optional. If a site that can utilize one association for both client-server directions (dual use) attempts to establish an association with a site that does not support dual use, it is the responsibility of the dual use site to fall back to single use associations.

An interruption on a data link, such as a failure of the physical link or the lower layers should result in an exception report from the lower protocol layers up to the client and server TASE.2 layers. This should cause the client or server to Abort.

Whenever an association is Concluded or Aborted, other associations already established should remain operational to minimize the impact of data exchanges over other associations or to other clients. An implementation of TASE.2 that takes down all associations in such a case is not following the spirit of the TASE.2 standards and should be avoided, even though it may still be in compliance with the protocol standards. This also applies during on-line configuration, where it should be possible to make changes in a database which affects only one remote control center with ICC-specific scope in the MMS domain without taking down other established associations to other control centers.

14.2 Local Implementation Setup Issues

When a utility implementing TASE.2 joins an existing network or begins communicating with another TASE.2 implementation, there are a number of issues that should be decided among the data exchange members.

– The maximum number of associations that will be allowed.
– The maximum exchange frequency of data should be agreed to in order to avoid overloading a SCADA/EMS with data requests.
– Which data types can be specified as critical data.
– The use and specification of retry counters.
– The assignment of values to the Information Reference Number used by most data objects.
– Which component structure will be used for the Indication Point objects. The use of Quality, Time Stamp, and Change-of-Value counter is optional, and not all vendors support all structures defined in the standards documents.
– The choice of STATE of DISCRETE for status points.
– The choice of periodic reporting of all Indication Point objects or RBE.
– The choice of reporting all or only the last change in an Indication Point object since the last information report where there are multiple changes since the last report.

14.3 Specific Conformance Block Issues

Individual conformance blocks in TASE.2 have specific user considerations. Some of these issues are local to the utility and some are issues that should be discussed with the TASE.2 and SCADA/EMS vendors prior to procuring a TASE.2 implementation.

14.3.1 Block 1 (Data Set Definition Management)

A concern among TASE.2 users is how to ensure synchronization of data set definitions at both the client and server sites.

14.3.1.1 Data Set Definition

The approach assumed by TASE.2 is for the client to create all data sets each time the association for transferring the data defined in data sets is established with another TASE.2 server. This would ensure that data set definitions are synchronized at least each time an association is restored after being brought down for whatever reason. This means that the TASE.2 server would not retain any data set definitions after an association with a remote client is brought down. The main drawback for sites with large amounts of data seems to be the time required to create all data sets before any data is actually transferred, but this approach must be used if interoperability is to be guaranteed.

If this approach is followed, then it is recommended that to provide more flexibility in handling the status of data set definitions after an Abort, the status of data set definitions after an interruption should be made a configurable parameter. This makes it possible to maintain the data set definitions and reuse them after an Abort and reestablishment of a new association.

A second, acceptable approach is for the server to *always* retain data set definitions whether or not associations exist with a TASE.2 client. Then it would be up to the client to periodically or on request verify the definitions at the server, perhaps using the Get Data Set Names and Get Data Set Element Names operations to compare the lists of Data Value object at the server with the lists known to the client. However, this requires that both the client and server expect to operate in this fashion ahead of time to avoid the possibility of client and server operating with different data set definitions.

In any case, since the client does not know what the server may have for data set definitions after an association has been terminated for some reason, the client should always verify that the definitions on the server match the client definitions each time an association is established. How this is done is a local implementation issue.

14.3.1.2 Data Set Updates

To detect a change in list of available Data Value objects at a server, TASE.2 provides a BLT version number that is checked by a client whenever an attempt to establish an association is made. If the BLT version number has changed, then the association will be concluded. There is no other protocol mechanism in TASE.2 to alert a client of such changes, nor is there a way to identify what changed, especially if new objects are added. If a Data Value object that a client had defined in a Data Set is now no longer available, then when an attempt to create that Data Set is made, an error message will be returned by the server indicating either that some Data Value is no longer accessible or no longer defined, but it will be up to the client to identify which object is affected.

Therefore, it is recommended that some scheme be defined outside TASE.2 to alert a client of specific changes. The simplest approach is to have a TASE.2 server site operator agree to email, telephone, or FAX notices of any changes affecting a client (i.e., addition, deletion, or modification of a point). Perhaps changes could be sent as alarms defined as pseudo-points with Blocks 1-2 or as an alarm message in Block 4.

14.3.2 Block 2 (Extended Data Set Condition Monitoring)

When using report-by-exception, there is always a small chance that due to either the client or the server system being down, or due to communications problems, the two databases will not be identical. The integrity scan (analogous to an RTU integrity scan) is used by TASE.2 to resynchronize databases. The use and frequency of integrity scans should be decided by data exchange members who have implemented Block 2. The agreed-to values would then

be specified as attributes in the Start DS Transfer Set request sent by the client to initiate data transfers.

When using report-by-exception for analog values, the use and specification of deadbands that reduce unnecessary transmission of minor changes should be decided by data exchange members.

An issue for discussion between the TASE.2 implementers and the utility is where the deadbands for analog and database state for status points will be monitored. Is this a SCADA/EMS or a TASE.2 implementation responsibility?

14.3.3 Block 4 (Information Messages)

14.3.3.1 Operator Messages

Block 4 can be used to send operator messages. After a TASE.2 implementation has received an operator message, it must be passed to the SCADA/EMS for presentation to the dispatchers or operators. How will the SCADA/EMS display, save, retrieve and purge the resulting message files?

14.3.3.2 Binary File Transfers

Block 4 allows for the transfer of small, binary files. These files will need to be stored in SCADA/EMS directories and the end user notified that they have been received. Will existing files automatically be overwritten? Will the local SCADA/EMS utilize the version information to automatically create new copies of the file? How will end users be notified? What convention will be used to identify the binary files as EXCEL, MS-Word, WordPerfect, etc.?

14.3.3.3 Requesting an Information Message Object

TASE.2 does not support the request of a specific information message or object.

However, when the need to support this exists, some utilities have solved this problem through establishing a convention for the use of the Information Reference number, which is a 32-bit integer. This number can be broken down into 9 bytes or fields. Each byte (or combination of 2 or more bytes) can be assigned a meaning. One byte can be reserved for indicating whether the information message is a request for a specific information message object or whether it is the actual object. For example, byte 4 could be encoded as follows:

1) Information Message data object.
2) Request for the Information Message object identified by the rest of the Information Reference number.

The rest of the Information Reference number would remain unchanged. The Info Stream attribute would be empty for the request.

14.3.3.4 Segmenting Long Information Messages

Information messages must fit within the maximum length MMS PDU, which is negotiated between client and server when an association is established within the MMS protocol layer. The typical maximum value is 32 000 bytes, but can be as small as 8 000 bytes. If messages longer than the negotiated value need to be transferred, then the use of a standard file transfer protocol, such as FTP or FTAM, is recommended. TASE.2 does not provide a capability to segment a long message into shorter segments and then reassemble the segments in the correct order at the client end.