Working Implementation Agreements for Open Systems Interconnection Protocols: Part 14 - Virtual Terminal

Output from the December 1993 Open Systems Environment Implementors' Workshop (OIW)

SIG Chair: **Michelle Conaway, HFSI**
SIG Editor: **Michelle Conaway, HFSI** Workshop Editor: **Brenda Gray**

Foreword

This part of the Working Implementation Agreements was prepared by the Virtual Terminal Special Interest Group (VTSIG) of the Open Systems Environment Implementors' Workshop (OIW). See Part 1 - Workshop Policies and Procedures in the "Draft Working Implementation Agreements Document" for the workshop charter.

Text in this part has been approved by the Plenary of the above-mentioned Workshop. This part replaces the previously existing chapter on this subject.

Only the pages that were changed in December 1992 are being printed. Please refer to the September 1992 Working Document for additional information.

Three normative annexes are given.

Future changes and additions to this version of these Implementor Agreements will be published as a new part. Deleted and replaced text will be shown as strikeouts. New and replacement text will be shown as shaded.

Table of Contents

Annex C (normative)

Annex D (informative)

Part 14 ISO Virtual Terminal Protocol

Editor's Note - References to Stable Agreements in this part refer to Version 5.

0 Introduction

See Stable Agreements.

1 Scope

1.1 Phase Ia agreements

See Stable Agreements.

1.2 Phase Ib agreements

See Stable Agreements regarding Forms profile.

1.3 Phase II agreements

See Stable Agreements regarding X.3 profile, Generalized Telnet profile and the S-mode Paged Application Profile.

1.4 Phase III agreements

Develop ISPs for A-mode Generalized Telnet profile, A-mode Transparent profile, S-mode Forms profile, S-mode Paged profile, and associated control objects.

Develop interoperability test cases for the Generalized Telnet profile.

Develop an ISP for Use of Directory by Vt entities.

Develop conformance tests for the Generalized Telnet profile.

2 Normative references

See Stable Agreements.

3 Status

These agreements are being done in phases. Below is the current status of each phase.

3.1 Status of phase Ia

The Phase Ia Agreements, which include the profiles for Telnet and Transparent operation, are complete and were stabilized in May, 1988. See Stable Agreements.

3.2 Status of phase Ib

The Forms profile of Phase 1b was stabilized in December, 1988. Alignment with EWOS Forms profile was achieved in September, 1989. See Stable Agreements.

3.3 Status of phase II

The S-mode Paged Application Profile is being progressed as PDISP 11187-2 (AVT-23 S-mode Paged Application Profile).

The X.3 profile was stabilized in December 15, 1989.

The Generalized Telnet profile was stabilized in December 13, 1991.

It is intended that Phase II agreements be compatible with Phase I agreements.

3.4 Status of phase III

Phase III is still in progress and includes the remaining work on the Generalized Telnet interoperability test cases, VT use of directory, and the Generalized Telnet conformance tests.

The S-mode Forms and S-mode Paged VTE profiles and their associated control objects have been submitted to SGFS. The A-mode Generalized Telnet and A-mode Transparent VTE profiles and their associated control objects have been approved by the regional workshops for submission to SGFS.

The S-mode Forms and Paged Application profiles and the A-mode Generalized Telnet and Transparent Application profiles are awaiting approval by the regional workshops.

It is intended that Phase III agreements be compatible with those of the previous phases.

4 Errata

See Stable Agreements.

5 Conformance

See Stable Agreements.

6 Protocol

See Stable Agreements.

7 OIW registered control objects

7.1 Sequenced Application (SA)

See Stable Agreements.

7.2 Unsequenced Application (UA)

See Stable Agreements.

7.3 Sequenced Terminal (ST)

See Stable Agreements.

7.4 Unsequenced Terminal (UT)

See Stable Agreements.

8 OIW defined VTE-profiles

8.1 Telnet profile

See Stable Agreements.

8.2 Transparent profile

See Stable Agreements.

8.3 Forms profile

See Stable Agreements.

8.4 X3 profile

See Stable Agreements.

8.5 Generalized Telnet profile

See Stable Agreements

8.6 S-mode Paged Application profile

See Stable Agreements.

Annex A (normative)

Specific ASE requirements

See Stable Agreements.

Annex B (normative)

Clarifications

See Stable Agreements.

Annex C (normative)

Object identifiers

See Stable Agreements for Object Identifiers assigned to objects in the Stable Agreements. Object Identifiers below have been assigned to objects for which work is still in progress.

General Identifiers:

```
oiw-vt-rep OBJECT IDENTIFIER ::= { oiw-vt repertoire(2) }

oiw-vt-font OBJECT IDENTIFIER ::= { oiw-vt font(3) }

oiw-vt-colour OBJECT IDENTIFIER ::= { oiw-vt colour(4) }

oiw-vt-directory OBJECT IDENTIFIER ::= { oiw-vt useOfDirectory(5) }
```

Annex D (informative)

Recommended practice_Operating X Window System over OSI upper layers

This annex provides a "recommended practice" for the operation of the X Window System (X) over an OSI upper layer stack. The "recommended practice" provides an interim[1] solution for an area not addressed by base standards or existing profiles. This recommended practice reflects OIW agreement.

It is recommended that this interim solution be used when mapping X over an OSI upper layer stack. However, implementors should note the following_future specifications of the regional workshops may possibly result in different solutions than those proposed in this recommended practice.

D.1 Background

X is a graphical user interface standard which enables a user to view and gain access to multiple computer applications from a single window or multiple windows on a display screen. X is based on a client/server architecture which allows applications and resources to be distributed across a network.

The **X server** is a software program that is resident on a user's display unit that acts as an intermediary between the user and applications running on a local or remote system. The X server also maintains complex data structures such as specific windows, cursors and fonts which can be referenced and utilized by applications. Input from the keyboard and/or mouse is collected by the X-server and passed to local and/or remote applications for processing.

Applications are referred to as **X clients**. These applications access the display unit by sending messages to the X server which is then able to perform two dimensional drawing of lines, shapes and text.

X products are based on a de facto standard (MIT-X) maintained by the MIT X Consortium. However, this specification does not provide for the operation of X over OSI-based networks.

Two OSI mapping specifications were created to define the operation of X over an OSI upper layer stack: EWOS Technical Guide 13 (ETG13) and part 4 of ANSI dpANS X.196 (X3.196). Parts 1-3 were intended to define the X protocol. Part 4 was based on ETG13. .X3.196 never progressed beyond the draft proposal stage. ETG13 was approved by EWOS in 05/91.

ETG13 explicitly defines:

a) the required OSI upper layer facilities;

b) the mapping of the OSI upper layer services for sending and receiving X protocol.

Since the creation of these documents, the ISO ISP 11188-3 *Common Upper Layer Requirements_Part 3: Minimal OSI upper layer requirements* (CULR-3) came into existence. CULR-3 defines the minimal set

[1] It is intended that this Recommended Practice will be progressed as an RWS technical report.

of OSI upper layer facilities for basic communications applications such as X.

Unlike ETG13, this specification does not itself specify the required upper layer facilities. Rather, it references CULR-3 to indicate the required OSI upper layer facilities. On the other hand, like ETG13, it specifies the mapping of X to the OSI upper layers services (ACSE, Presentation and Session). The mapping specified is compatible with that in ETG13.

This specification is intended to be as brief as possible. ETG13 includes additional guidance and explanatory material for implementors.

D.2 Mapping specification

This clause defines the mapping of the OSI ACSE (ISO 8649) and Presentation Layer (ISO 8822) services for sending and receiving X messages. This mapping uses the following ACSE and presentation services:

a) ASSOCIATE;

b) RELEASE;

c) ABORT;

d) A-P-ABORT;

e) P-DATA.

The required ACSE, presentation and supporting session facilities are discussed in clause D.3

For the purposes of this specification, the operation of X over the OSI upper layers is referred to as **X-osi**.

D.2.1 Summary of mapping

All the X protocol Request, Reply, Error and Event messages (i.e., the "X messages") use the encodings specified in MIT-X. The X messages are treated by this mapping as unstructured stream of octets. Any arbitrary sequence of consecutive octets can be treated as a single octet-aligned presentation data value this is transmitted as the user data on a Presentation P-DATA primitive. The OPEN DISPLAY Request and Reply messages are treated in the same way, and are carried on P-DATA. This mapping does not use the user data of the ACSE services.

The OSI upper layer stack supporting X-osi shall be mOSI compliant as defined in clause D.3.

D.2.2 Association establishment

The initiative for connection and association establishment is always with the X client. The X client establishes a new association with the desired X server by issuing an A-ASSOCIATE request. As part of the A-ASSOCIATE procedure, an OSI transport-connection is established to the X server system. The class of Transport protocol is out of scope of this specification. There is no requirement for X clients or X servers to re-use OSI Transport connections.

Once the transport-connection is established, an AARQ PDU carried in a Presentation Connect request (CP PPDU) that is in turn carried in a Session Connect request (CONNECT SPDU). The parameters shall include:

a) Application Context Name : This shall be the value "x-application context", defined in ETG13 and shown below:

b) Presentation Context Definition List : Shall include the ACSE presentation context and the X-osi presentation context, using the abstract and transfer syntax names defined in ETG13 and shown below. Other contexts may be offered (these may include synonyms or alternative names for X abstract or transfer syntax);

c) Presentation context identifiers shall be integers not greater than 255. This is a more severe restriction than ISO ISP 11188-1, Common Upper Layer Requirements_Part 1: Basic connection-oriented requirements (CULR-1), that permits 2-octet integers.

d) The user information field of the A-ASSOCIATE request shall be absent.

All other parameters are subject only to the requirements of mOSI compliance (see clause D.3).

If the X server accepts the association, the Application Context Name parameter on the A-ASSOCIATE response shall have the same value as that received on the indication. The ACSE and X-osi presentation contexts shall be accepted. If synonym abstract syntax or transfer syntax names for X-osi were offered and recognized, only one shall be accepted (i.e., following this exchange, there shall be a unique presentation context established for X-osi). The user information field of the A-ASSOCIATE response shall be absent.

D.2.3 Data exchange

As stated in the summary above, once the association is established, all X-messages are carried as user data on P-DATA primitives, each carrying a single PDV-list element containing a single "octet-aligned" presentation data value, which is some sequence of consecutive octets from one or more X-messages. No correlation is required between the PDVs (i.e. between successive P-DATAs) and the division between the X-messages : the division into PDVs is entirely at the sender’s option. (Obviously, in practice there will be some correlation, but there is no requirement to achieve this, nor should receivers rely on it.)

D.2.4 Connection termination

A CLOSE DISPLAY request from an X client is mapped to an A-RELEASE request. After receiving an A-RELEASE indication, the X server responds with an A-RELEASE response. Neither the request or response primitive shall contain any User Information.

A KILL CLIENT request from another client results in the issue of an A-ABORT request by the X server. A protocol or internal procedural error in either the X client or the X server also results in the issue of an A-ABORT request. The A-ABORT indication will contain the Abort Source parameter with the value "ACSE service-user".

The receipt of an A-ABORT indication with the Abort Source parameter having the value "ACSE service-provider" indicates a failure in either the local or peer ACSE. The receipt of an A-P-ABORT indication indicates a failure in the supporting Presentation Layer or below.

D.3 Required OSI upper layer facilities.

X is a basic communications application as defined in the CULR-3. That is, it simply requires the ability to open and close communications with a peer and to send and receive messages with the peer. The required facilities of the OSI upper layers (Session, Presentation, and ACSE) are specified by stating the minimal mOSI compliance requirements as defined in the CULR-3.

mOSI compliance requirements depend on whether a system supports one or more X clients (requests an association) or X servers (accepts an association request).

D.3.1 X client mOSI compliance

An upper layer stack that supports an X client shall be mOSI compliant category I or category II.

An X client stack has the following minimal compliance requirement based on Table 2 in the CULR-3.

a) "Establishment role" shall have the value "Initiator" or "Both". An X client is always the association initiator; it is never an association-responder.

b) "Normal data role" shall have the value "Both". An X client shall be able to send or receive data.

c) "Release role" shall have the value "Requestor", or "Both". A CLOSE DISPLAY request is mapped to A-RELEASE.

d) "Authentication" shall have the value "Supported" or "Not supported". The X client - X server association does not use the ACSE Authentication functional unit.

e) "AC negotiation" shall have the value "Supported" or "Not supported". The X client - X server association does not use the ACSE Application context negotiation functional unit.

f) "All "m" parms sent and received and CULR-1 compliance?" shall have the value "Yes". If the value is "Yes", the stack is mOSI compliant, category I or category II.

g) "All "o" parms sent and received?" shall have the value "Yes" or "No." If the value is "Yes", the stack is category I. If the value is "No", the stack is of category II. In this case, the X client stack is only required to support the following features for sending(see Table 3).

_Called AE title

_ Form1 (Directory name)

D.3.2 X server mOSI compliance

An upper layer stack that supports an X server shall be mOSI compliant category I or category II. The X server stack has the following compliance requirement based on Table 2 in the CULR-3.

a) "Establishment role" shall have the value "Responder" or "Both". An X server is always the association responder; it is never an association-initiator.

b) "Normal data role" shall have the value "Both". An X server shall be able to send or receive data.

c) "Release role" shall have the value "Acceptor", or "Both". The receipt of an A-RELEASE indication indicates a CLOSE DISPLAY request from the X client.

d) "Authentication" shall have the value "Supported" or "Not supported". The X client - X server association does not use the ACSE Authentication functional unit.

e) "AC negotiation" shall have the value "Supported" or "Not supported". The X client - X server association does not use the ACSE Application context negotiation functional unit.

f) "All "m" parms sent and received?" shall have the value "Yes". If the value is "Yes", the stack is mOSI compliant, category I or category II.

g) "All "o" parms sent and received?" shall have the value "Yes" or "No". If the value is "Yes", the stack is category I. If the value is "No", the stack is of category II. No category II features are required for sending.

D.4 Object identifiers

Object identifiers used for this specification are assigned in ETG13.[2]

Application context for X-osi :
{iso(1) identified-organization(3) ewos(16) eg(2) vt(7)
x-osi(10) application-context(1) }

Abstract syntax name:
{iso(1) identified-organization(3) ewos(16) eg(2) vt(7)
x-osi(10) abstract-syntax-version-1(2) }

Transfer syntax name:
{iso(1) identified-organization(3) ewos(16) eg(2) vt(7)
x-osi(10)
binary-transfer-syntax-version-1(3) }

2 These EWOS based object identifiers were also referenced in the last draft of X3.196_part 4.

D.5 Recommended encoding

It is recommended that the encoding of the Presentation PCI for the P-DATA follow a particular set of choices, among the optional features allowed by BER. This makes the P-DATA a (nearly) fixed header and allows implementations to be optimized to process this encoding. An implementation must be able to handle alternative encodings (i.e. any allowed by BER, subject to the restraints of CULR-1), within the mapping specification that each P-DATA carries a single octet-aligned presentation data value. The recommended encoding is :

a) the fully-encoded-data (SEQUENCE OF PDV-list) shall contain exactly one PDV-list;

b) both the SEQUENCE OF PDV-list and the PDV-list shall have indefinite length, but shall contain no levels of construction other than those required by the data types;

c) the length of the presentation-context-identifier value shall be expressed in short form;

d) the presentation-context-identifier value shall be encoded in one octet;

e) the OCTET STRING of presentation-data-values will contain a single presentation data value and shall have primitive encoding and

f) the (definite) length of this OCTET STRING shall be expressed in exactly four octets (i.e., the length itself will occupy three octets, prefixed by one octet which defines the length of this length).

These encoding choices mean that each TSDU user data consists of 16 octets of header, the X-message octets, and 4 octets of trailer (all zero). The length of the X-message segment is in the last three octets of the header.

This recommendation is identical to that in ETG13 except for the length field in (6). In ETG13 this is for a length of 1+4, not 1+3. This gives a 17-octet header. Since the X protocol, and many implementations go to some effort to get things on 4-byte boundaries, it is better to make this apply to X-osi as well. If a truly enormous P-DATA is needed i) the implementation is being very clever with its buffering; ii) it will have to use a longer length field; iii) the receiver is required to handle any legal encoding anyway.

D.6 Differences from ETG13

D.6.1 Abstract and transfer syntax names

In ETG13 the abstract and transfer syntax names are defined as names for the syntaxes defined in part II of X3.196, and ETG13 includes a copy of the April 1990 text for this. Since this is just a definition of the X data formats, there will be no problem in using them for X protocol as defined in MIT-X. ETG13 explicitly allows the extensibility features of X to be used without altering the syntax names.

Strictly speaking, X uses two transfer syntaxes, and the OPEN DISPLAY request defines which one will be used. The transfer syntax name defined in ETG13 covers both the "MSBfirst" and "LSBfirst" forms.

D.6.2 Application process title and application entity qualifier

ETG13 requires that the Called Application Process Title parameter on the A-ASSOCIATE request be a Directory Name (i.e. form1) in which the last RDN is the attribute value assertion CommonName=<displaynumber>, where <displaynumber> is a string representing the X Window System server number (and thus most commonly "0"), and that the Called Application Entity Qualifier be CommonName = "X-Window-System". The requirement was intended to facilitate X-osi : X-other relays, but this really requires integration with RFC 1275 to be general.

Although ETG13 requires these values it also recommends that implementations accept other values (or no value). Therefore there should be no interworking problems by omitting this requirement here.