

© British Standards Institution. No part of this publication may be photocopied or otherwise reproduced without the prior permission in writing of BSI

Draft for Development

Open systems interconnection: specification of the basic class protocol for job transfer and manipulation (ISO DP 8832)

Projet à développer. Interconnexion des systèmes ouverts: protocole de classe de base pour le transfert et la manipulation du travail — Spécifications

Entwurf zur weiteren Ausarbeitung. Verbindung offener Systeme; Protokoll der Basisklasse für Auftragsübermittlung und Manipulation

To avoid delay in preparing the content of this publication for printing, it was reproduced from the only available copy of the finally agreed document. The resultant poor print quality is very much regretted.

Contents	BS page	BS page
National foreword	2	18 Deferred processing of a work specification
Committees responsible	Back cover	19 Reference resolution
		20 Procedures for transmission
Draft for Development		21 Document movement procedures
0 Introduction	3	22 Report movement procedures
1 Scope and field of application	7	23 Work manipulation procedures
2 References	8	24 Action on J-TASKEND requests
3 Definitions	8	25 Task-end spawning procedures
4 Abbreviations	11	26 Termination report generation
5 Overview	11	
5.1 Model of JTM operation	11	Section three. Static conformance
5.2 Service primitives used	15	27 General requirements
5.3 Testing	16	28 Functional classes
5.4 Summary of JTM architecture	17	28.1 Support for Basic Class OSI-job submission
6 Conformance	19	28.2 Support for a Basic Class JTM monitor device
		28.3 Support for a Basic Class JTM display device
Section one. JTM datatypes		28.4 Basic Class JTM support for a file-store
7 Introduction to JTM datatype definition	20	28.5 Basic Class JTM support for a printer
8 Names	21	28.6 Basic Class JTM support for a jobmill
8.1 Global names	21	28.7 Basic Class JTM support for language xyz
8.2 Names local to an open system	23	28.8 Full Basic Class JTM support
8.3 Names local to a User-identification-authority	24	29 Management function requirements
8.4 Name lists	25	30 Documentation
8.5 Context names	25	
8.6 Document class names	26	Annexes
8.7 Presentation contexts for documents	27	A JTM Testing Procedures
9 Messages	28	A.1 Introduction
10 CCR and JTM diagnostics	28	A.2 Test configuration
11 Transfer elements	30	A.3 Management functions
11.1 Top-level fields	30	A.4 Loss of secure data
11.2 Audit elements	33	A.5 Initiation agency tests
11.3 Monitor specifications	34	A.6 Sink agency tests
11.4 Authorisation and permission elements	35	A.7 Execution agency test
11.5 Document movement operations	36	A.8 JTM activity tests
11.6 Work-manipulation operations	39	A.9 Programming language tests
11.7 Report-movement operation	40	A.10 Error cases
12 Work display documents	42	A.11 Masquerade checks
12.1 Semantic restrictions	43	B Parameters for transfer
13 Summary of datatypes	43	B.1 Connection and association establishment
		B.2 P-service parameters
Section two. JTM procedures		B.3 C-service parameters
14 Introduction to the procedures	50	C Management functions
14.1 General requirements	50	D The GraphicString datatype
15 Processing of J-INITIATE requests	51	D.1 Introduction
15.1 Creation of work specification	51	D.2 Graphic character strings
15.2 Authorisations	55	D.3 Encoding rules for graphic character strings
15.3 Permissions	57	D.4 User definition
15.4 JTM-action-parameters	57	
15.5 Proformas	57	National appendix
15.6 J-INITIATE confirm	57	A Commitment, concurrency
16 Receipt of a P-DATA indication	58	and recovery user data
17 Initial processing of a work specification	63	
		Figure
		1 JTM upper layer architecture

National foreword

This Draft for Development has been prepared under the direction of the Office and Information Standards Committee. The primary objective of this Draft for Development is to give advance guidance to product and systems designers concerned with job transfer and manipulation in the Application Layer of the Basic Reference Model for Open Systems Interconnection (OSI).

The text of the Draft for Development is identical with the first Draft Proposal ISO/DP 8832 issued November 1984 (reference ISO/TC 97/SC 21 N 211) by the International Organization for Standardization (ISO).

For ease of reproduction the ISO text of the first Draft Proposal has been reproduced unchanged.

A second Draft Proposal ISO/DP 8832.2 was issued July 1985 (reference ISO/TC 97/SC 21 N 521) which contained changes to make the Draft Proposal compatible with ISO/DIS 8650/3 (published by BSI as DD 110). It is not intended to reproduce the text of the second Draft Proposal, but national appendix A contains additional material to enable the current text to be used in conjunction with DD 110. This material gives practical guidance to implementors until the publication of ISO/DIS 8832 which is expected in early 1986.

The UK has actively participated through BSI Technical Committee OIS/121 in the development of the ISO Draft Proposal. The Technical Committee will review any agreed amendments or addenda to the ISO text and will, where appropriate, update the text of this Draft for Development via an amendment or revision. When the ISO draft reaches an appropriate level of agreement and stability, this Draft for Development may then be converted to a British Standard.

First-time readers of this Draft for Development are advised to read the tutorial annex (annex D of DD 105) of the job transfer and manipulation service before proceeding to the main text. They are also warned that this Draft for Development refers heavily to DD 110 dealing with the commitment, concurrency and recovery protocol and is not usable without that text.

This Draft for Development is closely related to DD 105 which defines the service provided by the protocol specified in this Draft for Development.

A brochure giving details of existing and projected British Standards publications for Open Systems Interconnection (OSI) may be obtained from the Sales Department, BSI, Linford Wood, Milton Keynes, MK14 6LE at no charge. Press releases for new OSI, data communications and computer graphics British Standards publications will be issued to the computing press, and details of all new publications will be recorded in full in *BSI News*.

Terminology and conventions. The text of the Draft Proposal has been approved as suitable for publication as a Draft for Development without deviation. Some terminology and certain conventions are not identical with those used in Drafts for Development; attention is drawn especially to the following.

Wherever the words 'International Standard' appear, referring to this Draft for Development, they should be read as 'Draft for Development'.

Cross-references

International standards	Corresponding British Standards
ISO 7498-1984	BS 6568 : 1984 Description of basic reference model for open systems interconnection (Identical)
ISO/DP 8831 (numbered ISO xxxx in the text)	DD 105 : 1985 Open Systems Interconnection: job transfer and manipulation concepts and services (Identical)
ISO/DIS 8649	DD 109 : 1985 Open Systems Interconnection: definition of common application service elements (CASE) : commitment, concurrency and recovery (Identical)
ISO/DP 8822 (numbered ISO pppp in the text)	DD 101 : 1985 Open Systems Interconnection: presentation service description (Identical)
ISO/DIS 8824 (numbered ISO aaaa in the text)	DD 103 : 1985 Open Systems Interconnection: specification for abstract syntax notation one (ASN.1) (Identical)
ISO/DIS 8825 (numbered ISO eeee in the text)	DD 104 : 1985 Open Systems Interconnection: basic encoding rules for abstract syntax notation one (ASN.1) (Identical)

At the time of publication of this Draft for Development, the references given in 2.2, 2.4 and 2.5 are at the Draft International Standard stage and the reference given in 2.3 is at the Draft Proposal stage.

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature because the subject matter is in the progress of ongoing development within the International Standards Committee (ISO/TC 97/SC 21 Information Retrieval, Transfer and Management for Open Systems Interconnection).

It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

A review of this Draft for Development will be carried out not later than two years after its publication. Notification of the start of the review period, with a request for the submission of comments from users of this Draft for Development will be made in an announcement in the appropriate issue of *BSI News*. Observations which it is felt should receive attention before the official call for comments will be welcomed.

According to the replies received, the responsible BSI Committee will judge whether the Draft for Development should be converted into a British Standard or what other action should be taken.



0 Introduction

This Standard specifies an Application Layer Protocol which supports the JTM Basic Class service elements defined in ISO xxxx (JTM Service Definition).

Application Layer Protocols support service elements which are referencable by other OSI Standards, but are more commonly used by vendor-specific implementations to support an interface to human beings, to devices or to general-purpose programming languages. In the latter cases, we speak of the service primitives being mapped by the implementation onto events in the real-world.

Standardisation of these mappings for certain specific devices or programming languages is not precluded, but is not currently being undertaken by ISO.

In order to support its service elements, an Application Layer Protocol may make use of either the Presentation Layer service elements, possibly enhanced by one or more sets of Common Application Service Elements, or may make use of the service elements supported by some other Application Layer Protocol.

The JTM protocol supports service elements which can be referenced by other Application Layer Standards, or which can be mapped by an implementor onto devices, human interfaces, or programming language interfaces. It uses the Presentation Service and CASE Kernel, together with the Commitment, Concurrency and Recovery (CCR) common application service elements, as defined in ISO 8649/3 (CCR Service definition) to provide this service.

The CCR procedures are also invoked for all activity by a JTM implementation when it accesses JTM agencies.

DD 106:1985

When an implementation receives an incoming P-DATA indication, it needs to determine that this communication is intended to invoke the procedures in this Standard. These procedures represent an application-context. This application-context is called JTM-with-CCR, and is selected by the sender of a JTM Transfer Element prior to its transfer. The way in which an application association is established between instances of two application layer entities, and the mechanisms for selecting the JTM-with-CCR application context, are not determined by this Standard. The service and protocol elements for performing these tasks are defined and specified in ISO 8649/2 (CASE Kernel Services) and ISO 8650/2 (CASE Kernel Protocol).

When two implementations of JTM-with-CCR communicate, there is also a need for agreement, by presentation layer negotiation, on

- a) the encoding rules to be applied to the datatypes whose abstract syntax is defined using the notation of ASN.1 in CCR and in SECTION ONE of this Standard; and
- b) the encoding rules to be applied to the datatypes defining the documents being transferred by JTM.

These agreements form the presentation-context for CCR and for JTM respectively, and are negotiated using the presentation service. A default set of encoding rules for the JTM context are specified in SECTION THREE of this Standard, and a default set for the CCR presentation context are specified in 8650/3 (CCR Protocol).

This protocol is a subset of the JTM Full Class Protocol in the following sense:

- a) if an implementation conforming to the JTM Full Class Standard is activated by a J-INITIATE service primitive or JTM PDU conforming to the Basic Class Subset, the resulting activity can be carried out with remote systems conforming

only to the Basic Class Protocol Standard; and

b) activity initiated on a Basic Class implementation can be carried out with remote systems conforming to the Full Class; and

c) if an implementation conforming to the JTM Full Class Standard is activated by a J-INITIATE service primitive or JTM PDU outside of the Basic Class Subset, and the resulting activity uses a remote system conforming only to the Basic Class Protocol Standard, then an appropriate diagnostic message results; and

d) an implementation conforming to the Full Class Standard also conforms to this Basic Class Standard; and

e) the JTM PDU defined in SECTION ONE of this Standard is a valid instance of the JTM PDU defined in the Full Class Standard.

SECTION ONE of this Standard specifies the abstract syntax of the JTM datatypes, using the notation specified in ISO aaaa (ASN.1).

SECTION TWO of this Standard specifies the procedures which are to be followed by an implementation when JTM service primitives are issued in an application-context which includes JTM-with-CCR. (Note that another application-context includes JTM-with-CCR if some other Standard, such as a management Standard, references the JTM service elements.) In this SECTION, reference has to be made to parts of a work specification. One possible abstract syntax for the semantics of most parts of a work specification is the "Transfer-element" datatype defined in SECTION ONE, and this syntactic representation is used in specifying parts of a work specification. The use of this notation does not in any way constrain the way in which a work specification (a semantic construct) is held locally: it is necessitated by the

DD 106:1985

inability to specify semantic values without the use of some syntactic notation.

SECTION THREE specifies the requirements on an implementation of JTM, and defines a set of terms which may be used by an implementor to describe a conforming JTM implementation. It also specifies default encoding rules by reference to the basic encoding rules of ASN.1 specified in ISO eeee.

Annex A is not part of the Standard. It outlines a set of test procedures which can be applied to a JTM implementation. It is expected that this Annex will form the basis for Standardisation of JTM test procedures, but is included in this Standard as an interim provision.

Annex B is part of the Standard. It specifies the values of the parameters of the presentation service and CCR primitives to be used by a JTM implementation for transfer of a Transfer Element.

Annex C is part of the Standard and specifies those local management functions which are referenced in the procedures of SECTION TWO, but whose detailed operation is not Standardised. It is expected that many implementations will make the values returned by such functions configurable by the user of the implementation; such provision is required (see clause 29) for a small number of the functions.

Annex D is part of the Standard and specifies an ASN.1 datatype which is widely used in JTM.

Temporary note This definition has been placed in an Annex to aid its migration to ASN.1 as UNIVERSAL 25.

1 Scope and field of application

This Standard specifies the behaviour which must be exhibited by an implementation claiming to conform to this JTM Basic Class Protocol Standard.

It specifies both dynamic conformance and static conformance for each of a set of terms which can be used to describe an implementation. It can be referenced by other OSI Standards (using the notation defined in the JTM Service Definition) to invoke the procedures specified in this Standard.

The Standard is to be applied by an implementor in developing a conforming implementation, and can also be referenced when the requirements for an implementation are specified.

The facilities provided by a JTM implementation are applicable to any field of activity in which asynchronous movement of documents is to occur.

The Standard does not completely determine the concrete transfer syntax to be used, but does specify a default value.

The Standard specifies the "application context requirement" name to reference the procedures of this Standard in application context negotiation (JTM-with-CCR application context).

The Standard specifies the "presentation contexts requirement" name to reference the abstract syntax of a JTM Transfer Element, specified in this Standard by use of ASN.1.

The Standard specifies the "concrete transfer syntax" name to reference the concrete transfer syntax obtained by applying the ASN.1 Basin Encoding Rules to the abstract transfer syntax specified using ASN.1.

DD 106:1985

2 References

- 2.1 ISO xxxx Information Processing -
Open Systems Interconnection -
JTM Service Definition

- 2.2 ISO 8649 Information Processing -
Open Systems Interconnection -
Definition of Common Application
Service Elements

- 2.3 ISO pppp Information Processing -
Open Systems Interconnection -
Presentation Service Definition

- 2.4 ISO aaaa Information processing - Open Systems
Interconnection - Specification of Abstract
Syntax Notation One (ASN.1)

- 2.5 ISO eeee Information processing - Open Systems
Interconnection - Basic Encoding Rules
for ASN.1.

3 Definitions

- 3.1 All definitions in ISO xxxx (JTM Service Definition) and in
ISO 8649/3 (CCR Service Definition) are used in this Standard.

- 3.2 static conformance: a statement of the requirement for sup-
port by an implementation of a valid set of features from among those
defined by the Standard.

- 3.3 dynamic conformance: a statement of the requirement for an
implementation to adhere to the behaviour prescribed by this Standard
in an instance of communication.

3.4 transfer element: the JTM PDU which is used to transfer the JTM information (semantics) called a work specification, between open systems.

Note A transfer element has a fully defined syntax, and is used to transfer the information (semantics) called a work specification between open systems. Within an open system the information (work specification) may be held in whatever way is considered appropriate locally. The procedures of this Standard are written throughout with reference to the fields of a Transfer-element datatype representing the work specification, in order to be precise.

3.5 initial processing: the procedures carried out by a JTM implementation before offering commitment to an atomic action which produced a work specification.

3.6 deferred processing: the procedures carried out by a JTM implementation on a work specification (held as secure data) after the implementation has taken responsibility for it.

3.7 primitive type: a name for a set of values.

3.8 abstract syntax (of the JTM PDU): a data type definition, produced using a repertoire of primitive types and means for combining them, that specifies the set of possible values of the PDU in a way which does not completely determine the representation of the PDU during transfer.

3.9 notation for abstract syntax definition: a set of rules for defining an abstract syntax.

3.10 oriented octet: an octet whose end-bits are named and distinguished, usually by the terms most significant bit and least significant bit.

DD 106:1985

3.11 concrete transfer syntax (of the JTM PDU): the representation of the JTM PDU during transfer, expressed as the value of a sequence of oriented octets.

Note The representation of a sequence of oriented octets on parallel or serial communication lines is specified by lower layer standards, and is not in the scope of JTM.

3.12 encoding rules: a set of rules, associated with a notation for abstract syntax definition, that enable a concrete transfer syntax to be derived for any PDU whose abstract syntax is specified using that notation.

3.13 application context name: a name which unambiguously references the (Standards defining the) complete set of abstract syntax definitions (and their semantics) to be used during the lifetime of a connection, and the procedures to be followed in sending or receiving them.

3.14 presentation context requirements name: a name which unambiguously references a set of abstract syntax definitions.

3.15 concrete transfer syntax name: a name which unambiguously references an encoding for a presentation context requirement.

3.16 local management functions: functions (whose internal details are not standardised) which return values which determine the value of certain work specification fields or which select between protocol options. (See Annex C for a complete list.)

3.17 human-readable text: text which clarifies a fully-defined diagnostic code, and which is expected to be read and understood by a human being.

Note The language in which the human-readable text is expressed is not defined in this Standard. An implementation may be configurable to produce text in any one of several languages, using different character sets. Fields of CCR primitives are used to specify the character set and hence indicate the preferred language; text is tagged with the identification of the character set used.

4 Abbreviations

4.1	JTM	Job Transfer and Manipulation
4.2	CCR	Commitment, Concurrency, and Recovery
4.3	CASE	Common Application Service Elements
4.4	ASN.1	Abstract Syntax Notation One
4.5	PDU	Protocol Data Unit

5 Overview

5.1 Model of JTM operation

5.1.1 JTM entities and agencies

The JTM service provider is represented by a collection of open systems. Each open system contains zero, one or more conceptual JTM agencies and a single JTM entity. (Reference to a single JTM (static) entity does not preclude the simultaneous operation of multiple (dynamic) instances of this entity to handle independent JTM transfers.) A real computer may contain several open systems (as seen by JTM), and a single open system (as seen by JTM) may consist of several real computers. The real-world equivalent of the JTM agencies is specified by the implementation, which also specifies the real-world

DD 106:1985

events (which may be geographically distributed) corresponding to the issue of JTM service primitives. In this way, the issue of JTM service primitives in an implementation becomes visible and testable.

Note Where the implementation conforms to some other Standard which references the JTM Standard, then the JTM service primitives may only be visible through effects defined by the other Standard.

The JTM entities jointly progress a work-specification (created as a result of J-INITIATE, of reporting, or of spawning). The responsibility for progressing a work specification always resides with exactly one JTM entity. This responsibility is passed by sending a "Transfer-element" (the work specification in a defined syntactic form) from one JTM entity to another as an atomic action.

A JTM entity having responsibility for a work specification holds enough information as secure data to completely define the work specification. Note that a work specification is a semantic construct, and may be held in a local system in any manner which is desired.

5.1.2 Use of the Presentation Service

The communication between the JTM entities uses a datatype called a "Transfer-element", whose abstract syntax and semantics is completely defined by this Standard. This is transferred using a concrete transfer syntax determined by the negotiated presentation context, with a default specified in this Standard.

The entire transfer element is transferred using a single P-DATA primitive. This primitive is issued in a suitable presentation context on an established presentation connection with the required properties. These properties and context requirements are specified in Annex B. The service primitives to establish a presentation con-

nection with these properties are defined in ISO pppp (Presentation Service Definition). Their use is not constrained further by this JTM Standard.

It should be noted that, apart from the use of CCR for ensuring reliability and returning diagnostics, the JTM transfer in Basic Class is of a connectionless, simplex, nature. It involves a single P-DATA, in one direction only.

5.1.3 Use of the CASE Kernel

The CASE Kernel is used to negotiate the use of JTM-with-CCR on a new or existing presentation connection, and to select this application context.

The service primitives to perform this operation are specified in ISO 8649/2. Their use is not constrained further by this JTM Standard.

Local scheduling requirements may cause abandonment of the JTM-with-CCR application-context (or of the underlying presentation connection). Later recovery (resynchronisation) is performed by use of C-RESTART within a new JTM-with-CCR application context established between the same instances of the same entity types.

5.1.4 Relation to CCR

The JTM entity uses CCR for communication to

- a) other JTM entities, using the Presentation Service with CCR service elements; and
- b) JTM agencies, using the J-service primitives which include CCR semantics.

DD 106:1985

At any time, at most one of these communications is to the JTM entity's CCR superior, and the rest are to its CCR subordinates. The CCR procedures (defined in terms of all the CCR primitives received or issued) are applied to the entire activity of the JTM entity.

The application of the CCR procedures, through the use of the CCR Common Application Service Elements, ensures reliable transfer of responsibility for work specifications and consistency of operation.

The CCR procedures also provide for the return of diagnostic information when initial processing fails. (Following initial processing, diagnostic information on failure is returned by the JTM reporting mechanisms.)

Finally, the CCR procedures provide for the (optional) transmission of information which can be used by an implementation to decide when to retry an attempt to send a transfer-element to another JTM entity, or when to abandon attempts to progress a received transfer-element during initial processing, and to revert to commitment at a lower commitment level.

This Standard does not require a JTM entity to provide this CCR timer information to a remote entity, so that an entity must be prepared to adopt a suitable local algorithm in the absence of information.

Note A suitable algorithm would be to attempt a transfer at exponentially increasing intervals of time over a period of several days, then to consider the attempt to have failed and invoke the JTM error handling procedures.

5.2 Service primitives used

The service primitives which invoke JTM procedures are:

- a) P-DATA indication in a JTM-with-CCR application and JTM presentation context;
- b) J-INITIATE-WORK request;
- c) J-INITIATE-WORK-MAN request;
- d) J-TASKEND request;
- e) all CCR indications and confirms in the JTM-with-CCR application context.

The service primitives invoked by JTM procedures are:

- a) P-DATA request in a JTM-with-CCR application and JTM presentation context;
- b) J-DISPOSE indication;
- c) J-GIVE indication and response;
- d) J-STATUS indication;
- e) J-KILL indication;
- f) all CCR requests and responses in the JTM-with-CCR application context.

In all the above cases, these primitives are issued as part of a CCR commitment group.

DD 106:1985

5.3 Testing

5.3.1 The lower tester

Provided all applicable Standards are known, it is possible to attach suitable test equipment to the physical medium, and to use it to determine the PDUs which are being transmitted, and to generate PDUs. Where the lower layers of the JTM implementation under test are considered to be conforming (they have already been tested by the tests specified for their layer), and the test equipment contains a conforming implementation of the lower layers, it can be used to recognise or to generate conceptual events at the lower boundary of the layer under test (P-DATA indications in JTM context and C-service indications and confirms).

Where there is (unusually) a downwards facing visible interface below the layer under test, but above the physical medium, this provides an alternative means of recognising or generating conceptual events at the lower boundary of the layer.

Finally, events at the lower boundary of the layer under test may also be recognised or generated by exercising a corresponding visible interface in remote equipment. (Issuing P-DATA or C-service requests).

The equipment or set of procedures used to generate and display these events (using any of the above configurations) forms the lower tester. The detailed specification of this equipment or set of procedures is outside the scope of this Standard.

It is possible to apply a number of tests using the lower tester alone. Such tests can be independent of the details of the implementation, and are therefore generally to be preferred to tests using other interfaces. In general, however, tests at this interface alone are insufficient to detect certain forms of non-conformance.

5.3.2 The upper tester

An implementation of JTM either

- a) provides a visible interface for J-service primitives;
or
- b) specifies a referencing Standard which makes use of J-service primitives, and itself provides a visible interface.

A suitable test configuration or procedures can exercise this visible interface, and is necessary to perform any tests concerned with "real" services provided by JTM, such as the successful printing of output. This configuration or set of procedures forms the upper tester; it always contains some features which are dependent on the implementation, as Standards do not exist for interfaces (except for the physical medium).

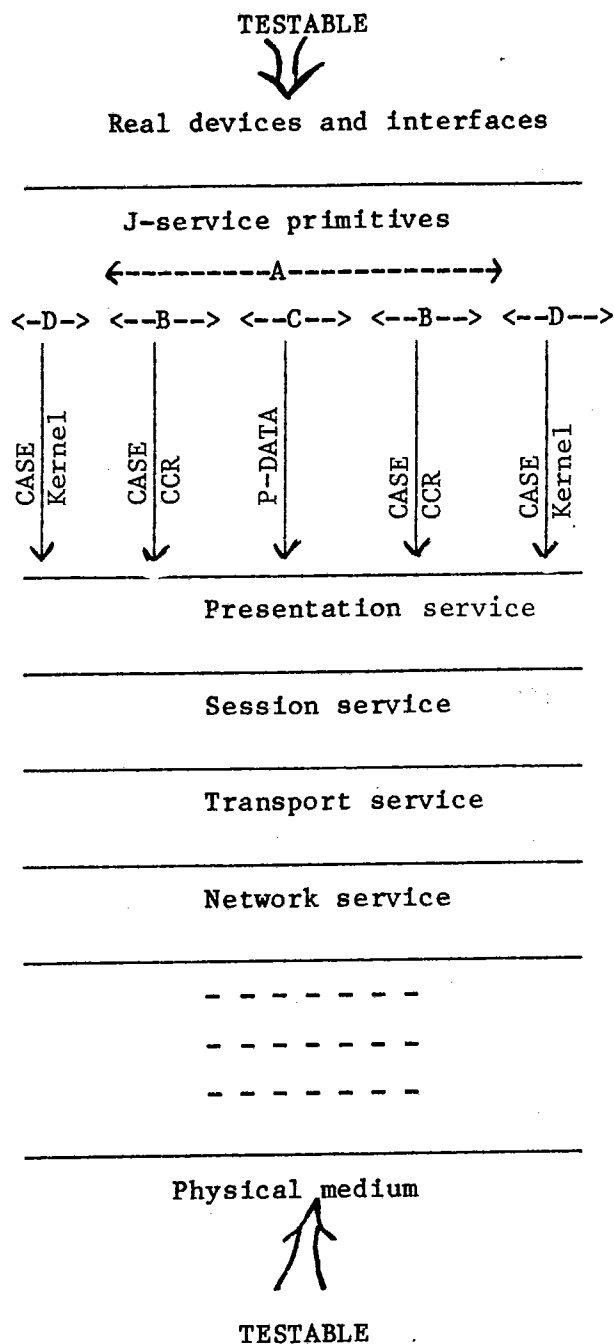
5.4 Summary of JTM architecture

The JTM architecture is given in Figure 1.

The figure reflects the fact that the CCR procedures are applied to coordinate and constrain events at the J-service and the Presentation-service boundary (in an interlinked way).

Note that in JTM-with-CCR application context

- a) P-service primitives in CCR presentation context are visible only as C-service primitives; and
- b) P-service primitives in JTM presentation context are visible as P-DATA; and



- A - JTM-with-CCR application context
- B - CCR presentation context
- C - JTM presentation context
- D - CASE Kernel presentation context

Figure 1 - JTM upper layer architecture

c) P-service primitives in CASE Kernel presentation context are not visible, but serve to establish or terminate the JTM-with-CCR application context.

6 Conformance

6.1 Clause 5 above was provided as background. There are no conformance requirements to clause 5. Where clause 5 is in disagreement with other clauses, the other clauses take precedence.

6.2 Dynamic conformance is specified in SECTION TWO using the datatype definitions given in SECTION ONE.

6.3 Static conformance is specified in SECTION THREE.

DD 106:1985

SECTION ONE - JTM Datatypes

7 Introduction to JTM datatype definition

This SECTION defines the JTM datatypes. It uses the standard notation of ISO aaaa (ASN.1) to define the abstract syntax of the JTM PDU - the Transfer Element datatype.

This clause specifies only the abstract syntax, and does not reference the encoding rules for generation of the concrete transfer syntax.

Clause 8 defines the datatypes used for naming. These appear throughout the later definitions.

Clause 9 defines the datatypes used for strings, messages, and diagnostics. These use the ASN.1 GraphicString datatype defined in Annex D.

Clause 10 defines the datatypes "Document" and "Document-class-name". These form an important part of later definitions, and represent "user data" in the JTM service.

Clause 11 defines a datatype called "Transfer-element", the JTM PDU. The JTM service is provided by generating, transferring and processing work specifications. Clause 11 defines the complete data structure for all CHOICES which are used in the Basic Class, giving the abstract syntax, and the semantic restrictions during transfer. The way instances of this datatype shall be created and the way they shall be processed on receipt is specified in Section Two.

Clause 12 defines the datatype "Work-display-doc". This is generated by the JTM service provider as specified in Section Two, and is the only JTM-defined document used in the Basic Class.

Clause 13 is a repetition of the datatype definitions in clauses 8, 9, 10, 11 and 12, and is intended for machine processing. If differences occur, clauses 8, 9, 10, 11, and 12 are authoritative.

Annex D defines the GraphicString datatype used throughout these definitions.

The definitions in this Section are related to those of the Full-Class Standard as follows:

- a) where the Full-Class Standard specifies other distinguished values for an INTEGER or a BITSTRING, these are not listed in this Standard;
- b) where the Full-Class Standard specifies other elements of a SEQUENCE or SET, these are specified in the Full-Class Standard as OPTIONAL, with a DEFAULT which is compatible with the Basic Class semantics; the Full-Class Standard specifies that where an element has the DEFAULT value, it shall not be present in the concrete transfer syntax;
- c) where the Full-Class Standard specifies other alternatives for a CHOICE, these are not listed in this Standard, and where this reduces the alternatives to a single one, the alternative is listed without the constructor CHOICE.

8 Names

8.1 Global names

JTM DEFINITION ::=

BEGIN

Open-system-name ::= ANY

User-identification-authority ::= ANY

DD 106:1985

END

Note

It is expected that an ISO Standard will be produced defining a data-type and transfer syntax for these names, establishing mechanisms for their allocation, and providing data for the directory functions defined below.

Instances of these datatypes are required to exhibit the following properties on all open systems communicating using this Standard:

- a) given an instance of an "Open-system-name" to be used for a JTM transfer, local directory functions are used to generate a unique set of addressing information to enable JTM transfers to be made to an open system; this open system is said to be the open system corresponding to the name; this correspondence is independent of the open system making the connection;
- b) the addressing information provided by an incoming call may be sufficient to determine that the call was made by a named open system;
- c) an instance of a "User-identification-authority" corresponds to a single source of user identifications among the communicating systems; a central or distributed authority ensures that this unambiguity of naming is maintained;
- d) there is no requirement for values of these two datatypes to be distinct; where they are not, the same value can be used for both an "Open-system-name" and for a "User-identification-authority"; where this is convenient for human administrations, and where the above requirements can be met.

For enterprise-specific use prior to an ISO Standard and name allocation scheme, the following datatypes shall be used:

JTM DEFINITIONS ::=

BEGIN

Open-system-name ::= [PRIVATE 1] GraphicString

User-identification-authority ::= [PRIVATE 2] GraphicString

END

Enterprise-specific mechanisms are used for unambiguous allocation of names and for distributing directory information in accordance with the above requirements.

Note Until such time as this clause is amended, an implementation claiming conformance to this Standard shall use the [PRIVATE 1] and [PRIVATE 2] datatypes.

8.2 Names local to an open system

JTM DEFINITIONS ::=

BEGIN

Agency-name ::= GraphicString

END

Each JTM execution agency, sink agency and source agency is associated with at least one instance of the datatype Agency-name. An instance of this datatype is associated with only one agency on a specific open system.

An open system uses directory functions to enable it to issue J-service primitives to a named agency. The directory information identifies whether the agency is a source, sink or execution agency.

DD 106:1985

8.3 Names local to a User-identification-authority

```
JTM DEFINITIONS ::=
BEGIN
  User-identification ::= GraphicString
END
```

The authorisation of activity invoked through JTM is based on the authentication of user identifications. A "User-identification" is used, within the scope of a "User-identification-authority", to identify a set of allowed activities.

Each open system is configured to recognise one or more instances of the type "User-identification-authority" (typically one). It contains local directory information which enables it to determine the allowed activities for some or all of the "User-identifications" issued by these "User-identification-authorities".

A "User-identification-authority" may be known only to a single open system. In this case all associated directories are local to that open system. Where a particular "User-identification-authority" is known to more than one open system, some directory information is needed at each open system, and protocols for remote password checking may be in use.

Tutorial note

The user identifications and the associated directories for authentication and authorisation are usually the normal mechanisms provided on most computer systems. The additional requirements for networking are the allocation of unambiguous names for user identification authorities, and arrangements for distributing the directories when the same user identification is to be used on more than one open system.

8.4 Name lists

```
JTM DEFINITIONS ::=
BEGIN
Name-list ::= SEQUENCE OF GraphicString
END
```

8.5 Context names

This clause specifies the names to be used for

- application context requirements
- presentation context requirements
- concrete transfer syntax

as required by ISO 8649/2 (Case Kernel) and by ISO pppp (Presentation Service).

The encoding of these names in fields of Case Kernel and Presentation PDUs is specified in those Standards.

8.5.1 Application context requirements

Communications activity proceeding in accordance with the requirements of this Standard shall be informally described as "JTM-with-CCR, Basic Class", and shall have the application context requirements name of

ISOxxxx-JTM-BASIC

8.5.2 Presentation context requirements

The datatypes "Transfer-Element" (see 11.1) and "Work-display-doc" (see clause 12) shall be sent and received in a presentation context defined by use of the presentation context requirements name of

DD 106:1985

ISOxxxx-JTM-BASIC

See also 8.7.

8.5.3 Concrete transfer syntax

When negotiating a concrete transfer syntax for the presentation context requirement of 8.5.2, the name

ISOxxxx-JTM-BASIC-1

shall be used to reference that concrete transfer syntax which is obtained by applying ISO eeee (the Basic ASN.1 Encoding Rules) to the datatypes defined in this Section.

No other concrete transfer syntax names are defined in this Standard. Other syntaxes may be defined and named by the mechanisms established by the transfer syntax Registration Authority.

8.6 Document class names

```
JTM DEFINITIONS ::=
BEGIN
Document-class ::= CHOICE
    {private  [0] GraphicString,
     JTM      [1] JTM-doc-class,
     standard [2] ANY          }
END
```

Values of the datatype for "standard" document classes are determined by the Document Class Registration Authority.

Note This CHOICE shall not be used until the authority is established.

Values of the GraphicString for "private" document classes are determined on an enterprise-specific basis. Prior to establishment of an ISO Registration Authority, implementations are recommended to use the names

JTM-TEXT
JTM-PRINT
JTM-BINARY

to reference the documents defined in Entries 1 to 3 of Annex C of ISO yyyy (JTM Service).

JTM DEFINITION ::=

BEGIN

JTM-doc-class ::= CHOICE

{work-display [Ø] NULL}

END

This Document-class references the document consisting of a single instance of the datatype "Work-display-doc" defined in clause 12.

8.7 Presentation contexts for documents

Each document transmitted by JTM has its own presentation context, which is not, in general, the JTM context.

The "Work-display-doc" is transmitted in the JTM presentation context specified in 8.5.2 with the encoding of 8.5.3.

Pending the establishment of a Document Class Registration Authority, documents

JTM-TEXT
JTM-PRINT
JTM-BINARY

DD 106:1985

shall also be transmitted in the JTM presentation context.

Note Clauses in this Standard referencing these documents are subject to amendment when the Register is established.

9 Messages

Messages are used to return diagnostic information to human users.

```
JTM DEFINITIONS ::=
BEGIN
Message ::= SEQUENCE OF GraphicString
END
```

It is intended that each "GraphicString" in a "Message" should be presented to humans on a separate line. The "GraphicString" shall not exceed 40 characters when used in a "Message". The total length of the "Message" is not limited.

Note The datatype "GraphicString" is defined in Annex D.

The graphics repertoires used in the "Message" datatype shall consist only of those listed in one of the "code-specifiers" of the CCR "diagnostic code indicator" parameter for the atomic action, or of characters from the International Reference Version of ISO 646.

Note Where the total work involves several atomic actions, the service provider securely retains the CCR diagnostic code parameter for use on subsequent atomic actions.

10. CCR and JTM diagnostics

JTM specifies a datatype for the "code" in a CCR and JTM diagnostic. The specification is

JTM DEFINITIONS ::=

BEGIN

JTM-code ::= SEQUENCE

```

    {jtm-user-code CHOICE
        {absent [0] NULL,
         user-defined [1] ANY }
    CHOICE
    {sf-not-supported [0] NULL,
     ue-unauthorised-access [1] NULL,
     ue-unauthorised-reporting [2] NULL,
     ue-no-document [3] NULL,
     ue-agency-unknown [4] NULL,
     ue-no-agency-document [5] NULL,
     ue-no-authorisation-for-transfer [6] NULL,
     ue-site-unknown [7] NULL,
     w-document-name-changed [8] NULL,
     ua-killed-by-manipulation [9] NULL}}

```

END

The "jtm-user-code" shall be the CHOICE NULL unless this Standard is referenced by another ISO Standard which specifies the datatype.

Note Section Two specifies the circumstances in which each of these error codes is generated. Those marked "sf" are service failures, "ue" are user errors, "me" are management errors, "ua" are user actions, and "w" are warnings.

When this datatype is carried by CCR (referenced as type ANY in CCR), it shall be encoded according to the encoding rules for the JTM presentation context, tagged with the context as specified in the presentation protocol, then the resulting octets placed in the CCR PDU as the encoding of ANY.

DD 106:1985

11 Transfer elements

This clause defines a syntactic structure, and specifies semantic restrictions for a "Transfer-element". This is the JTM Protocol Data Unit. It also serves to define the contents of a JTM work specification (a semantic construct with no defined abstract syntax).

A JTM transfer, as specified in SECTION TWO, consists of a single P-DATA primitive (within a CCR atomic action), carrying

- a) a value of the "Transfer-element" datatype (in JTM presentation context); followed by
- b) zero one or more values of a document datatype, each of which may be from different presentation contexts.

11.1 Top-level fields

JTM DEFINITIONS ::=

BEGIN

Transfer-element ::= [APPLICATION 0] SEQUENCE

osi-job-submission-system	[0]	Open-system-name	,
initiating-identification	[1]	Identification	,
initiating-time	[2]	Time-stamp	,
osi-job-name	[3]	GraphicString	,
osi-job-local-reference	[4]	GraphicString	,
sub-job-name-list	[5]	Subjob-name-list	,
audit-trace	[6]	SEQUENCE OF Audit-element,	
primary-osi-job-monitor	[7]	Monitor-specification	,
authorisations	[8]	SET OF	
		Authorisation-element	,
permissions	[9]	SET OF	
		Permission-element	,
		COMPONENTS OF	
		Subjob-specification	}

```

Subjob-name-list ::= SEQUENCE OF SEQUENCE
                    {proforma-name GraphicString,
                     instance      INTEGER      }

```

```

Subjob-specification ::= SEQUENCE
{target      [10] Open-system-name      ,
 type        [11] Subjob-type           ,
 actions     [12] JTM-action-parameters,
 proforma-list [13] SEQUENCE OF Proforma }

```

```

Subjob-type ::= CHOICE
{document-movement [0] NULL,
 work-manipulation [1] NULL,
 report-movement   [2] NULL}

```

```

JTM-action-parameters ::= CHOICE
{document movement [0] Document-movement-operations,
 work-manipulation [1] Work-manipulation-operations,
 report-movement   [2] Report-movement-operation   }

```

```

Proforma ::= SEQUENCE
{proforma-name [0] GraphicString      ,
 COMPONENTS OF
 Subjob-specification}

```

END

The following data-types are defined in later sections:

Identification	11.4
Time-stamp	11.5
Audit-element	11.2
Monitor-specification	11.3
Authorisation-element	11.4
Permission-element	11.4

DD 106:1985

Document-movement-operations	11.5
Work-manipulation-operations	11.6
Report-movement-operation	11.7

A work specification also contains

```
<time waiting = I>
<estimated size = I>
<agency activity parameters>
```

These values are not transferred between open systems in a transfer element, so corresponding fields are not provided.

It also contains

```
<CCR parameters>
```

recording the value used for the CCR "diagnostic code" parameter.

Finally, it also contains zero, one or more documents which are transferred as additional values on the P-DATA primitive.

When a work specification is awaiting a J-TASKEND, the outermost "Subjob-specification" has a null "target", "type", and "actions".

11.1.1 Semantic restrictions

The following restrictions apply during the transfer of a "Transfer-element":

- a) the value of the "osi-job-submission-system Open-system-name" shall equal the value of the first "Open-system-name" in the "audit-trace" (see 11.2);

- b) the "instance" in the "subjob-name-list shall be greater than or equal to one;
- c) the "Subjob-type" and the "JTM-action-parameters" shall be one of the following pairs:
- | | |
|-------------------|-------------------------------|
| document-movement | Document-movement-operations; |
| work-manipulation | Work-manipulation-operations; |
| report-movement | Report-movement-operation; |
- d) the "Subjob-type" in a "Subjob-specification" in a "Proforma" shall not be "report-movement";
- e) if the "Subjob-type" in the "Transfer-element" is "report-movement", the "proforma-list" shall be "{}", and there shall be no further datatype values in the P-DATA.
- f) the elements of "authorisations" and "permissions" shall all have distinct values.
- g) the top-level "proforma-list" shall be either "{}" or a sequence containing a single "Proforma".
- h) if a "Proforma" is present in the "Transfer-element", there shall be no "embedded" choice in any "Document-pointer" in the proforma (see 11.5);
- j) the P-DATA shall contain at most one value (document) following the Transfer-element;

11.2 Audit elements

JTM DEFINITIONS ::=

BEGIN

Audit-element ::= SEQUENCE

DD 106:1985

```

{sender [0] Open-system-name,
 status [1] CHOICE
   {unknown      [0] NULL,
    known        [1] NULL,
    authenticated [2] NULL} }

```

END

11.2.1 Semantic restrictions

During transfer, the value of the last "Audit-element" shall have a "status" of "unknown". The setting and use of other "Audit-element" values is specified in Section Two.

11.3 Monitor specifications

JTM DEFINITIONS ::=

BEGIN

Monitor-specification ::= SEQUENCE

```

{report-selector      [0] BITSTRING
 {manipulation-termination (0),
  abnormal-termination    (1)} ,
 monitor                [1] Open-system-name,
 disposal-instructions [2] SEQUENCE
 {agency [0] Se-identification ,
  doc-name [1] Document-se-reference} }

```

END

The following data-types are defined in later sections:

Se-identification	11.5
Document-se-reference	11.5

11.3.1 Semantic restrictions

The following restrictions shall apply in the Basic Class:

- a) the "report-selector" shall be

'11'B

- b) the value of the "se-location Open-system-name" in the "Se-identification" (see 11.5) shall equal the value of the "monitor Open-system-name".

11.4 Authorisation and permission elements

JTM DEFINITIONS ::=

BEGIN

Authorisation-element ::= SEQUENCE

```
{id          [0] Identification  ,
 validation [1] CHOICE
    {checked-index [0] INTEGER ,
      password     [1] CHOICE
        {unset [0] NULL      ,
          set   [1] OCTETSTRING} } }
```

Permission-element ::= Identification

Identification ::= CHOICE

```
{open-system [0] Open-system-name      ,
 authority   [1] User-identification-authority ,
 user        [2] SEQUENCE
    {authority [0] User-identification-authority ,
      id       [1] User-identification      } }
```

END

11.4.1 Semantic restrictions

The checked-index shall be a non-zero positive integer, and shall not contain a value greater than the number of "Audit-element" fields in the "audit-trace", numbering the first such field as "1".

DD 106:1985

11.5 Document movement operations

JTM DEFINITIONS ::=

BEGIN

Document-movement-operations ::= SEQUENCE OF SEQUENCE

```

{type          [0] Document-class  ,
 ses           [1] SEQUENCE OF
               Se-identification,
 document-block [2] Single-form    }
```

Se-identification ::= SEQUENCE

```

{location [0] Open-system-name,
 agency   [1] Agency-name     }
```

Single-form ::= SEQUENCE

```

{doc-name [0] Document-se-reference ,
 docs     [1] SEQUENCE OF Document-pointer}
```

Document-se-reference ::= SEQUENCE

```

{document-name [0] Name-list}
```

Document-pointer ::= CHOICE

```

{embedded          [0] NULL ,
 single-document-reference [1] SEQUENCE
  {action-open-system [0] Open-system-name ,
   source              [1] Source-identification ,
   document-name       [2] Name-list ,
   state               [3] CHOICE
    {not-attempted [0] NULL} } }
```

Note If the Transfer-element represents a work-specification for which reference resolution has not yet been attempted, the "state" is "not-attempted". Reference resolution either turns the reference into an "embedded" document-pointer, or turns the "state" to "failed".

```

Source-identification ::= CHOICE
    { provider [0] NULL
      ,
      agency [1] SEQUENCE
        { location [0] Open-system-name,
          name [1] Agency-name } }

C-diagnostic-information ::= SET OF SET
    { generated-by [0] Open-system-name,
      code [1] JTM-code
      ,
      text [2] Message }

Time-stamp ::= CHOICE { not-available [0] NULL,
                        time [1] Time }

END

```

11.5.1 Semantic restrictions

The number of occurrences of the "embedded" CHOICE for "Document-pointer" shall exactly equal the number of values in the P-DATA following the Transfer-element. Each value is a document.

The class of each of these documents shall agree with the "Document-class" given in the "Document-movement" containing the "embedded" CHOICE of "Document-pointer". The first document corresponds to the first transmitted "embedded" CHOICE, the second to the next, and so on.

The "Document-movement-operations" shall consist of exactly one "document-movement" and the "ses" shall consist of exactly one "Source-identification".

The "SEQUENCE OF Document-pointer" in the "Single-form" shall be a single "Document-pointer". The "action-open-system" in the "single-document-reference" and the "location" in the "source-identification" shall be the receiving open system.

DD 106:1985

The single "document-movement" shall have the following generic values:

```
{type      t
ses        {{location      y,
               agency      z}},
document-block {doc-name{document-name d,
               doc      {embedded NULL} }}
```

where t is an instance of "Document-class" set to

```
private "JTM-TEXT"
or private "JTM-PRINT"
or JTM work-display NULL
```

y is an instance of "Open-system-name" with a value equal to the value of the instance of "Open-system-name" in the "target", z is an instance of a GraphicString used by the receiving open system as the name of a sink or execution agency, and d is any value of the type "Name-list".

11.5.1.1 In a work-manipulation transfer-element proforma

The "document-movement" in a "Proforma" which is in a "Transfer-element" with "JTM-action-parameters" which are "Work-manipulation-operations" shall have the value:

```
{type JTM work-display NULL
ses {{location      y,
               agency      z}},
document-block {doc-name{document-name d
               docs {single-document-reference
                     {action-open-system w ,
                      source {provider NULL},
                      document-name {s }}}} }
```

where y is any "Open-system-name", z is any "GraphicString", d is any "Name-list", w is equal to the "Open-system-name" of the top-level "target" in the "Transfer-element" and s is a "GraphicString" equal to the "docname" in the "Work-manipulation-operations" (see 11.6).

11.5.1.2 In a document-movement transfer-element proforma

The "document-movement" in a "Proforma" which is in a "Transfer-element" with "JTM-action-parameters" which are "Document-movement-operations" shall have any value.

11.6 Work-manipulation operations

JTM DEFINITIONS ::=

BEGIN

Work-manipulation-operations ::= SEQUENCE OF
Work-operation

Work-operation ::= CHOICE

```
{select      [0] Selector      ,
 kill        [1] NULL          ,
 stop        [2] NULL          ,
 brief-display [3] GraphicString}
```

Selector ::= SEQUENCE

```
{selector-form [0] CHOICE
 {first-header-is [0] NULL},
 [1] Header-test }
```

Header-test ::= CHOICE

```
{      [0] Field-test      ,
 and-clause [1] SEQUENCE
 {header-one [0] Header-test,
 header-two [1] Header-test}}
```

DD 106:1985

Field-test ::= CHOICE

```

    {osi-job-submission-system-equals [0] Open-system-name,
      osi-jobname-equals                [1] GraphicString  }

```

END

11.6.1 Semantic restrictions

The "Work-manipulation-operations" shall be one of:

```

    {select      x ,
      kill      NULL}

```

or

```

    {select      x ,
      stop      NULL}

```

or

```

    {select      x,
      brief-display y}

```

where y is any "GraphicString", and x is an instance of "Selector" with the value

```

    {selector-form first-header-is NULL
      and-clause
        {header-one osi-job-submission-system equals p,
          header-two osi-job-name equals q          } }

```

where p is any instance of an Open-system-name, and q is any instance of a GraphicString.

11.7 Report-movement operation

JTM DEFINITIONS ::=

BEGIN

Report-movement-operation ::= SET OF

Single-report

Single-report ::= SEQUENCE

```

{reported-by           [0] Open-system-name,
  time                 [1] Time-stamp      ,
  osi-job-submission-system [2] Open-system-name,
  initiating-identification [3] Identification ,
  initiating-time        [4] Time-stamp      ,
  osi-job-name           [5] GraphicString  ,
  osi-job-local-reference [6] GraphicString  ,
  subjob-name-list       [7] Subjob-name-list,
  type                   [8] Subjob-type     ,
  event                  [9] CHOICE
  {manipulation-termination [0] SEQUENCE
    {spawn-count [0] INTEGER ,
      text       [1] Message } ,
    abnormal-termination [1] SEQUENCE
      {spawn-count [0] INTEGER ,
        text       [1] Message ,
        errors     [2] Diagnostic} }

```

Diagnostic ::= SEQUENCE

```

{originator           [0] CHOICE
  {source-details     [0] SEQUENCE
    {agency           [0] Source-identification,
      doc-name        [1] Name-list           },
    se-details        [1] SEQUENCE
      {agency          [0] Se-identification ,
        doc-name       [1] Name-list           },
    transfer-details   [2] Open-system-name },
  message             [1] C-Diagnostic-information ,
  when                [2] Time-stamp             }

```

END

DD 106:1985

11.7.1 Semantic restrictions

The "Subjob-name-list" shall satisfy the restrictions of 11.1.1(b).

The "spawn-count" fields in the "event CHOICE" shall all be greater than or equal to zero.

12 Work display documents

JTM DEFINITIONS ::=

BEGIN

Work-display-doc ::= [APPLICATION 4] SEQUENCE OF
SEQUENCE

{displayed-by [0] Open-system-name,
time [1] Time-stamp ,
display [2] SEQUENCE OF
Brief-display }

Brief-display ::= SEQUENCE

{details [0] SEQUENCE

{osi-job-submission-system [0] Open-system-name,
initiating-identification [1] Identification ,
initiating-time [2] Time-stamp ,
osi-job-name [3] GraphicString ,
osi-job-local-reference [4] GraphicString ,
subjob-name-list [5] Subjob-name-list,
type [6] Subjob-type },

status [1] SET OF

Destination-status }

Destination-status ::= SEQUENCE

{destination [0] CHOICE

{recipient [0] Open-system-name,
source-agency [1] SEQUENCE
{location [0] Open-system-name ,