

INTERNATIONAL STANDARD

**ISO
8832**

First edition
1989-06-01

Information processing systems — Open Systems Interconnection — Specification of the Basic Class Protocol for Job Transfer and Manipulation

*Systèmes de traitement de l'information — Interconnexion de systèmes ouverts —
Spécifications du protocole de base pour le transfert et manipulation de travaux*



Reference number
ISO 8832 : 1989 (E)

AVA Documents
Document Sharing Hub

ISO 8832 : 1989 (E)

Contents

Foreword	vi
Introduction	vii
Section 1 : General	1
1.1 Scope	1
1.2 Normative references	1
1.3 Definitions	2
1.4 Abbreviations	2
1.5 Relation of JTM to other services	2
1.5.1 JTM architecture	2
1.5.2 JTM ASEs and agencies	3
1.5.3 Use of the Presentation Service	3
1.5.4 Use of the ACSE ASE	3
1.5.5 Use of the CCR ASE	3
1.5.6 Service primitives referenced by this International Standard	4
1.5.7 Summary of JTM architecture	4
1.6 Conformance	4
Section 2 : JTM datatypes	5
2.1 Introduction to JTM datatype definitions	5
2.2 Names and messages	5
2.2.1 Global names	5
2.2.2 Names local to a JTM ASE	5
2.2.3 Names local to a User-identification-authority	5
2.2.4 Name lists	6

© ISO 1989

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

International Organization for Standardization

Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

2.2.5	Context names	6
2.2.6	Human-readable messages	7
2.3	Diagnostics	7
2.3.1	JTM Diagnostic codes	7
2.3.2	CCR diagnostics	7
2.4	User data on CCR primitives	7
2.4.1	User data on C-BEGIN request and indication	8
2.4.2	User data on C-READY request and indication	8
2.4.3	User data on C-REFUSE request and indication	8
2.4.4	User data on C-PREPARE request and indication	8
2.4.5	User data on C-RESTART request and indication	8
2.4.6	User data on C-RESTART response and confirm	8
2.5	Transfer elements	8
2.5.1	Top-level fields	9
2.5.2	Audit elements	10
2.5.3	Monitoring specifications	10
2.5.4	Authorisation and permission elements	10
2.5.5	Document movement operations	10
2.5.6	Work-manipulation operations	11
2.5.7	Report-movement operation	12
2.6	Work-display and report-display documents	12
2.7	Summary of datatypes	13
Section 3 : JTM procedures		16
3.1	Introduction to the procedures	16
3.1.1	General requirements	16
3.2	Processing of J-INITIATE requests	16
3.2.1	Creation of work specifications	16
3.2.2	Authorisations	17
3.2.3	Permissions	17
3.2.4	JTM-action-parameters	17
3.2.5	Proformas	17
3.2.6	J-INITIATE confirm	18
3.3	Procedures for receipt of work specifications	18
3.4	Initial processing of a work specification	19
3.5	Deferred processing of a work specification	20
3.6	Reference resolution	21

ISO 8832 : 1989 (E)

3.7	Procedures for transmission of work specifications	22
3.8	Document movement procedures	22
3.9	Report movement procedures	24
3.10	Work manipulation procedures	24
3.11	Action on J-END-SIGNAL requests	27
3.12	Spawning procedures	27
3.13	ABNORMAL-TERMINATION report generation	28
3.14	NORMAL-TERMINATION report generation	28
3.15	Action on J-MESSAGE requests	29
Section 4 : JTM functionality		30
4.1	Static conformance requirements	30
4.2	Functional classes	31
4.2.1	Support for Basic Class OSI-job submission	31
4.2.2	Support for Basic Class JTM monitoring	31
4.2.3	Support for Basic Class JTM manipulation	31
4.2.4	Basic Class JTM support for a local file-store	31
4.2.5	Basic Class JTM support for an output device	32
4.2.6	Basic Class JTM support for job processing	32
4.2.7	Basic class JTM support from language xyz	32
4.2.8	Full Basic Class JTM Support	32
4.3	Management function requirements	32
4.4	Documentation	33
Section 5 : JTM transfers		34
5.1	Application association control	34
5.1.1	Notation	34
5.1.2	Procedures	34
5.1.3	ASSOcnf-return codes	34
5.2	Parameters in Association Control Service Primitives	37
5.2.1	Parameters in A-ASSOCIATE service primitives	37
5.2.2	Parameters in A-RELEASE service primitives	37
5.3	P-service parameters	37
5.3.1	Parameter of P-DATA	37
5.3.2	Interruption of P-DATA	37
5.4	C-service parameters	40
5.4.1	C-BEGIN and J-BEGIN request and indication	40
5.4.2	C-READY and J-READY request and indication	40

5.4.3	C-REFUSE and J-REFUSE request and indication	40
5.4.4	C-RESTART and J-RESTART	41
Annexes		
A	Management functions	42
B	Document Types	45
B.1	Simple text document type	45
B.2	Simple print document type	47
B.3	Simple binary document type	49
B.4	Work-display document type	50
B.5	Report-display document type	51
C	JTM Testing Procedures	52
D	Summary of ASN.1 OBJECT IDENTIFIER assignments	56
E	Tutorial examples of protocol sequences	57

ISO 8832 : 1989 (E)**Foreword**

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8832 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Annexes A and B form an integral part of this International Standard. Annexes C, D and E are for information only.

Introduction

This International Standard specifies the behaviour of an Application Service Element which supports the Job Transfer and Manipulation (JTM) Basic Class Service defined in ISO 8831.

Application Service Elements support service primitives which are referencable by other OSI Standards, but are more commonly used by vendor-specific implementations to 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 primitives, an Application Service Element makes use of the Presentation Service, possibly enhanced by the use of one or more Common Application Service Elements, or of the service primitives supported by some other Application Service Element.

The JTM Application Service Element supports service primitives which can be referenced by other Application Service Elements, or which can be mapped by an implementor onto devices, human interfaces, or programming language interfaces. It uses the Presentation Service defined in ISO 8822, the Association Control Application Service Element defined in 8649, and the Commitment, Concurrency and Recovery (CCR) Common Application Service Element defined in ISO 9804 to provide the JTM service.

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

When an implementation receives an incoming P-DATA indication, it needs to determine that this communication is intended to invoke the procedures in this International Standard. These procedures represent an **application-context**. This application-context is called "ISO JTM Basic Class Application-context", and is established prior to the transfer of a JTM Transfer Element, using the services of ISO 8649.

When two implementations of "ISO JTM Basic Class Application-context" communicate, there is a need for agreement, by presentation layer negotiation, on

- the encoding rules to be applied to the datatypes whose abstract syntax is defined (using the notation of ASN.1) in CCR;
- the encoding rules to be applied to the datatypes whose abstract syntax is defined using ASN.1 in section 2 of this standard; and
- the encoding rules to be applied to the datatypes forming the documents being transferred by JTM.

These agreements form the **presentation-contexts** for CCR, for JTM, and for document transfer. They are negotiated using the presentation service. A mandatory set of encoding rules (to be supported by all implementations) for the JTM context are specified in section 5 of this International Standard, and a mandatory set for the CCR presentation-context are specified in ISO 9805. The mandatory set for documents are specified with the definition of the document type in annex B.

ISO 8832 : 1989 (E)

Section 2 of this International Standard specifies the abstract syntax of the JTM datatypes, using the notation specified in ISO 8824.

Section 3 of this International Standard specifies the procedures which are to be followed by an implementation when JTM service primitives are issued in an application-context which includes "ISO JTM Basic Class Application-context". (Another application-context includes "ISO JTM Basic Class Application-context" if some other standard references the JTM services.)

Section 4 specifies the requirements on an implementation of Basic Class JTM, and defines a set of terms which can be used by an implementor to describe a conforming Basic Class JTM implementation. It also specifies mandatory encoding rules by reference to the basic encoding rules of ASN.1 specified in ISO 8825.

Section 5 specifies the establishment and release of an application association which is to be used to support a JTM transfer, and the parameters of primitives used to obtain such an association and conduct a JTM transfer.

Annex A is part of this International Standard and specifies those local management functions which are referenced in the procedures of section 3, 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 4.3) for a small number of the functions.

Annex B is part of this International Standard and defines a number of document types which general-purpose Basic Class JTM implementations are expected to support.

Annex C is not part of this International Standard. It outlines a set of test procedures which can be applied to a Basic Class JTM implementation. This annex is expected to form the basis for standardisation of JTM test procedures, but is included in this International Standard as an interim provision.

Annex D is not part of this International Standard and summarises the assignment of ASN.1 OBJECT IDENTIFIER and ObjectDescriptor values.

Annex E is not part of this International Standard and provides tutorial examples of some protocol sequences.

Information processing systems — Open Systems Interconnection — Specification of the Basic Class Protocol for Job Transfer and Manipulation

Section 1 : General

1.1 Scope

This International Standard specifies the behaviour to be exhibited by an implementation claiming to conform to this International Standard.

It specifies both dynamic conformance and static conformance. It can be referenced by other OSI Standards (using the notation defined in the JTM Service Definition) to invoke the procedures specified in this International Standard.

This International 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.

This International Standard does not completely determine the transfer syntax to be used in an instance of communication, but does specify a transfer syntax which all implementations are required to support.

This International Standard specifies

- the application-context name to be used to reference the procedures of this International Standard in application-context negotiation;
- the abstract syntax name to be used to reference the abstract syntax of a JTM Transfer Element, CCR user data and JTM-defined documents, specified in this International Standard by use of ASN.1;
- the transfer syntax name to be used to reference the transfer syntax obtained by applying the ASN.1 Basic Encoding Rules to the abstract syntax specified using ASN.1.

This International Standard defines a set of local management functions which are needed in order to support the operation of a JTM implementation. These local management functions are in-

voked by the JTM Application Service Element. They are not modelled as operations within the application entity, nor do they form part of the normal services provided by or assumed by the JTM Application Service Element. They form a means of specifying the degree of flexibility which is permitted to or required of implementations conforming to this International Standard.

1.2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8571-3:1988, *Information processing systems - Open Systems Interconnection - File Transfer, Access and Management - Part 3 : File Service definition.*

ISO 8649:1988, *Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element.*

ISO 8650:1988, *Information processing systems - Open Systems Interconnection - Protocol specification for the Association Control Service Element.*

ISO 8822:1988, *Information processing systems - Open Systems Interconnection - Connection oriented presentation service definition.*

ISO 8824:1987, *Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).*

ISO 8825:1987, *Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*



Documents

Document Sharing Hub

ISO 8832 : 1989 (E)

ISO 8831:1989, *Information processing systems - Open Systems Interconnection - Job transfer and manipulation concepts and services*.

ISO 9804:—¹⁾, *Information processing systems - Open Systems Interconnection - Service definition for the Commitment, Concurrency and Recovery service element*.

ISO 9805:—¹⁾, *Information processing systems - Open Systems Interconnection - Protocol specification for the Commitment, Concurrency and Recovery service element*.

1.3 Definitions

For the purposes of this International Standard, the definitions given in ISO 8831 and ISO 8649 and the following definitions apply.

1.3.1 static conformance: a statement of the requirement for support by an implementation of a valid set of features from among those defined by this International Standard.

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

1.3.3 transfer element: part of the JTM PDU which is used to transfer the JTM information (semantics) contained in a work specification between open systems.

1.3.4 initial processing: the procedures carried out by a JTM implementation before offering commitment to an atomic action which generated a work specification at the open system.

1.3.5 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.

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

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

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

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

1.3.10 transfer syntax (of the JTM transfer element or of a document): the representation of the JTM transfer element or document 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.

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

1.3.12 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 an application association, and the procedures to be followed in sending or receiving them.

1.3.13 abstract syntax name: name which unambiguously references the abstract syntax of an identified set of datatype definitions.

1.3.14 transfer syntax name: a name which unambiguously references an encoding for an abstract syntax.

1.3.15 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 A for a complete list.)

1.3.16 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 International 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.

1.3.17 JTM Application Service Element: an abstract representation of those parts of an open system which perform the procedures specified in this International Standard.

1.4 Abbreviations

ASN.1	Abstract Syntax Notation One
CCR	Commitment, Concurrency, and Recovery
JTM	Job Transfer and Manipulation
PDU	Protocol Data Unit
ASE	Application Service Element
ACSE	Association Control Service Element

1.5 Relation of JTM to other services

1.5.1 JTM architecture

This International Standard specifies the operation of the Basic Class JTM Application Service Element.

This ASE can operate as part of an application-entity which contains only the Basic Class JTM ASE, the CCR ASE, and the ACSE ASE. It can provide its services directly to the User Element.

The JTM agencies are a conceptual model of the structures and operation of the User Element.

1) To be published

The Basic Class JTM ASE can also operate as part of an application entity which contains other ASEs which make use of the JTM services. It is not visible to the Basic Class JTM ASE whether it is providing services directly to the user element or to some other ASE.

This International Standard specifies static conformance requirements for the situation where the Basic Class JTM ASE provides services directly to the user element. Where the Basic Class JTM ASE is used by some other ASE, the static conformance requirements are specified by the standard specifying the ASE which is using the JTM services.

1.5.2 JTM ASEs and agencies

The JTM service provider is represented by a collection of JTM ASEs in application-entities. Each application-entity contains zero, one or more conceptual JTM agencies and a single JTM ASE. (Reference to a single JTM (static) ASE does not preclude the simultaneous operation of multiple (dynamic) instances of this ASE to handle independent JTM transfers.) A real computer system may support several application-entities with a JTM ASE, and a single application entity may model the operation of several real computers. The real-world equivalent of the JTM agencies is specified by the implementation, which also specifies the real-world 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 this International Standard, then the JTM service primitives might only be visible through effects defined by the other standard.

The JTM ASEs 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 ASE. This responsibility is passed by sending a "Transfer-element" (the work specification in a defined syntactic form) from one JTM ASE to another as an **atomic action**.

A JTM ASE 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.

1.5.3 Use of the Presentation Service

The communication between the JTM ASEs uses a datatype called a "Transfer-element", whose abstract syntax and semantics is completely defined by this International Standard. This is transferred using a transfer syntax determined by presentation layer negotiation.

The entire transfer element, together with any associated documents, is transferred using a single P-DATA primitive. This primitive is issued on an application association. The service primitives to establish and use an application association are defined in ISO 8649 and in ISO 8822 and their use is specified in section 5.

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 simplex nature. It involves a single P-DATA, in one direction only.

1.5.4 Use of the ACSE ASE

The ACSE ASE is used to begin and end the use of an application association in "ISO JTM Basic Class Application-context".

The service primitives to perform this operation are specified in ISO 8649 and their use is specified in section 5.

Local scheduling requirements may cause abandonment of the "ISO JTM Basic Class Application-context" (or of the underlying presentation connection). Later recovery (resynchronisation) is performed by use of C-RESTART within a new "ISO JTM Basic Class Application-context" established between the same application-entities.

On completion of JTM activity on an association, the association is relinquished by use of ACSE ASE primitives.

1.5.5 Use of the CCR ASE

The JTM ASE uses the CCR ASE for communication to

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

At any time, at most one of these communications is to the JTM ASE'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 ASE.

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

The CCR primitives 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 primitives 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 International Standard does not require a JTM ASE to provide this timer information to a remote JTM ASE, so that a JTM ASE has to 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.

ISO 8832 : 1989 (E)

1.5.6 Service primitives referenced by this International Standard

The service primitives which invoke JTM procedures are

- a) a P-DATA indication in the "ISO JTM Basic Class Application-context" carrying a data value in the JTM presentation-context;
- b) a J-INITIATE-WORK request;
- c) a J-INITIATE-WORK-MAN request;
- d) a J-END-SIGNAL request;
- e) a J-MESSAGE request;
- f) all CCR indications and confirms in the "ISO JTM Basic Class Application-context";
- g) A-ASSOCIATE, A-P-ABORT, A-ABORT, and A-RELEASE indications and confirms (see 3.3.4).

The service primitives invoked by JTM procedures are

- a) a P-DATA request in "ISO JTM Basic Class Application-context";
- b) a J-DISPOSE indication and response;
- c) a J-GIVE indication and response;
- d) a J-STATUS indication;
- e) a J-KILL indication;
- f) a J-STOP indication;
- g) all CCR requests and responses;
- h) A-ASSOCIATE, A-ABORT, and A-RELEASE requests and responses (see 3.3.4).

1.5.7 Summary of JTM architecture

The JTM architecture is illustrated in figure 1.

Note that in "ISO JTM Basic Class Application-context"

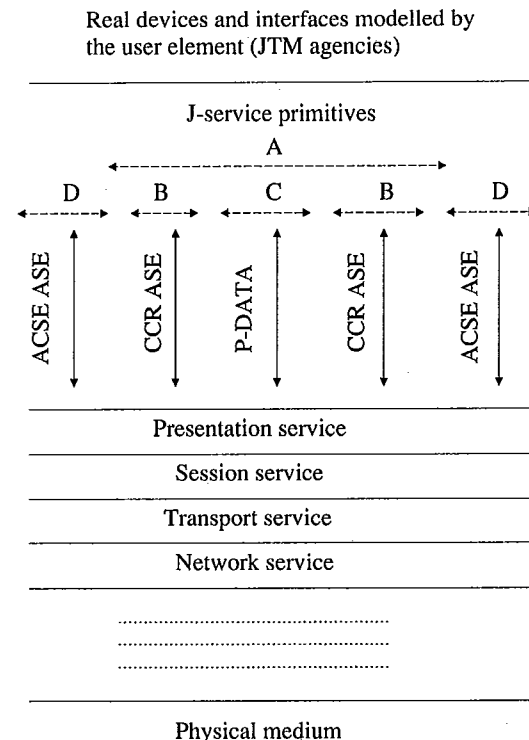
- a) P-service primitives carrying data values in CCR presentation context are visible only as C-service primitives; and
- b) P-service primitives carrying data values in JTM and document presentation contexts are visible as P-DATA; and
- c) P-service primitives carrying data values in ACSE ASE presentation context serve to establish or terminate the "ISO JTM Basic Class Application-context".

1.6 Conformance

1.6.1 Clause 1.5 above was provided as background. There are no conformance requirements to clause 1.5. Where clause 1.5 is in disagreement with other clauses, the other clauses take precedence.

1.6.2 Dynamic conformance is specified in section 3 and section 5 using the datatype definitions given in section 2.

1.6.3 Static conformance is specified in section 4.



- A - "ISO JTM Basic Class Application-context"
- B - CCR presentation-context
- C - JTM and document presentation-contexts
- D - ACSE presentation-context

Figure 1 — JTM upper layer architecture

Section 2 : JTM datatypes

2.1 Introduction to JTM datatype definitions

This section defines the JTM datatypes. It uses the standard notation of ISO 8824 (ASN.1) to define the abstract syntax of the Transfer Element datatype which forms part or all of the JTM PDU. (The rest of the PDU, if any, is a document whose abstract syntax is defined elsewhere.)

It also defines the abstract syntax of the datatypes which form the user data parameters of CCR primitives used by JTM.

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

Clause 2.2 defines the datatypes used for naming (these appear throughout the later definitions), and for human readable messages.

Clause 2.3 defines the datatypes used for diagnostics and for accounting information.

Clause 2.4 defines the datatypes used for the user data parameters of CCR primitives.

Clause 2.5 defines a datatype called "Transfer-element". The JTM service is provided by generating, transferring and processing work specifications. A Basic Class work specification (a semantic concept) is represented during transfer as a "Transfer-Element", together with zero or one documents. Clause 2.5 defines the complete "Transfer-Element" 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 3.

Clause 2.6 defines the datatypes "Work-display-doc" and "Report-display-doc". These are generated by the JTM service provider as specified in section 3, and are the only JTM-defined documents used in the Basic Class.

Clause 2.7 is a repetition of the datatype definitions in clauses 2.2, 2.3, 2.4, 2.5 and 2.6, and is intended for machine processing. If differences occur, clauses 2.2, 2.3, 2.4, 2.5, and 2.6 are authoritative.

2.2 Names and messages

2.2.1 Global names

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
JTM-name ::= Application-entity-title

User-identification-authority ::= Application-entity-title

Application-entity-title ::=
    ISO8650-ACSE-1.ApplicationTitle
END
```

NOTES

1 The above module reference uses the definition of the syntax of application-entity-titles provided in ISO 8650. The datatype and corresponding datavalues are imported into the JTM abstract syntax definition, that is to say, the values of these datatypes are transmitted in the presentation-context established for JTM PDUs.

2 Application-entity-titles are unambiguous and are supported by directory functions relating them to addressing information.

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

- a) given an instance of a "JTM-name" to be used for a JTM transfer, local directory functions are used to generate addressing information to enable JTM transfers to be made to a JTM ASE; this JTM ASE is said to correspond to the name; this correspondence is independent of the JTM ASE making the transfer;
- b) the addressing information provided by an incoming call may be sufficient to determine or to verify that the call was made by a particular JTM ASE;
- 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; the same value can be used for both a "JTM-name" and for a "User-identification-authority", where this is convenient for human administrations.

2.2.2 Names local to a JTM ASE

```
ISO8832-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 accessed by a specific JTM ASE.

A JTM ASE uses local directory functions to enable it to issue J-service primitives to a named agency. The local directory information identifies whether the agency is a source, sink or execution agency.

2.2.3 Names local to a User-identification-authority

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
User-identification ::= GraphicString
END
```

ISO 8832 : 1989 (E)

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

Each JTM ASE 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.

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 directory information when the same user identification is to be used on more than one open system.

2.2.4 Name lists

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Name-list ::= SEQUENCE OF GraphicString
END
```

The "Name-list" datatype is used for the identification of documents passed to and from JTM source, sink, and execution agencies.

2.2.5 Context names

This sub-clause specifies the names to be used for

- a) the names of the application-contexts which are used in A-ASSOCIATE service primitives in order to establish the use of an association for the procedures specified in this International Standard; and
- b) the name of the abstract syntax for all the datatypes defined in this International Standard and summarised in clause 2.7; and
- c) the name of the transfer syntax for these datatypes which is obtained by applying the Basic Encoding Rules (see ISO 8825) of ASN.1.

Specification of these names uses the following ASN.1 value definition:

```
jtm OBJECT IDENTIFIER ::= { iso standard 8832 }
```

The associated ObjectDescriptor value is

```
"ISO JTM Basic Class Standard"
```

NOTE — The ASN.1 definition of ObjectDescriptor and OBJECT IDENTIFIER permits, but does not require, the translation of ObjectDescriptor values and of identifiers in OBJECT IDENTIFIER values when the standard defining such values is translated into another language.

2.2.5.1 Application contexts

A Basic Class implementation is aware of two application-context-names. The first identifies the procedures of this International Standard and has the following ASN.1 "OBJECT IDENTIFIER" and "ObjectDescriptor" values:

```
{ jtm application-context (1) basic (1) }
"ISO JTM Basic Class Application-context"
```

The second identifies the full JTM procedures and has the following ASN.1 values:

```
{ jtm application-context (1) full (2) }
"ISO JTM Full Application-context"
```

2.2.5.2 Abstract syntax

Presentation data values received and transmitted by Basic Class systems (other than those specified by the ACSE and CCR ASEs) are either:

- a) values of the ASN.1 datatype "Transfer-element" defined in 2.5; or
- b) values of the ASN.1 datatypes

```
C-BEGIN-USER-DATA
C-READY-USER-DATA
C-REFUSE-USER-DATA
C-RESTART-RI-USER-DATA
C-RESTART-RC-USER-DATA
```

defined in 2.4; or

- c) presentation data values specified in a document type definition.

Presentation data values of type c) shall be transferred in the presentation context(s) established from the abstract syntax name(s) specified in the document type definition. (See annex B for document types defined in this International Standard.)

Presentation data values of types a) and b) shall be transferred in a presentation context established using the abstract syntax identified by the following ASN.1 "OBJECT IDENTIFIER" and "ObjectDescriptor" values:

```
{ jtm abstract-syntax (2) }
"ISO JTM Abstract Syntax"
```

NOTE — The document type definitions for "work-display" and "report-display" documents also use this abstract syntax name.

2.2.5.3 Transfer syntax

When negotiating a transfer syntax for the abstract syntax "ISO JTM Abstract Syntax" specified in 2.2.5.2, the following ASN.1 definitions shall be used to reference that transfer syntax which is obtained by applying ISO 8825 (the Basic ASN.1 Encoding Rules) to the datatypes defined in this section:

```
{ jtm transfer-syntax (3) }
"ISO JTM Transfer Syntax"
```

No other transfer syntaxes are defined in this International Standard. Other transfer syntaxes for the "ISO JTM Abstract Syntax" can be defined and named by other organisations, and may be used in presentation context negotiation. (See, however, 2.2.5.3.)

2.2.6 Human-readable messages

Messages are used to return diagnostic information to human users.

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Message ::= SEQUENCE OF GraphicString
END
```

The sequence shall consist of one or more elements, and each "GraphicString" shall consist of from zero to 40 characters.

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

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. (See 2.4.)

NOTES

- 1 The requirement to support ISO 646 is subject to variation in national standards which are otherwise equivalent to this International Standard.
- 2 Where the total work involves several atomic actions, the service provider securely retains the CCR diagnostic code parameter for use on subsequent atomic actions.

2.3 Diagnostics

2.3.1 JTM Diagnostic codes

JTM specifies a datatype for the “code” in CCR diagnostics. The specification is

ISO8832-JTM DEFINITIONS ::=	
BEGIN	
JTM-code ::=	SEQUENCE
{jtm-user-code	CHOICE
{absent	[0] NULL
user-defined	[1] EXTERNAL ,
	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
ue-disposal-error	[8] NULL
w-document-name-changed	[9] NULL
ua-killed-by-manipulation	[10] NULL
sf-protocol-error	[11] NULL
sf-transmission-failure	[12] NULL
sf-too-large	[13] NULL
rl-incoming-concurrency-limit	[14] NULL
rl-timeout	[15] NULL
rl-transmission-attempt	[16] NULL
rl-access-to-agency-document	[17] NULL
rl-document-disposal	[18] NULL
rl-manipulation-in-progress	[19] NULL
rl-agency-concurrency-limit	[20] NULL

```

rl-transmission-concurrency-limit [21] NULL
rl-internally-busy [22] NULL
sf-repeated-retries [23] NULL
ue-monitor-site-unknown [24] NULL
sf-incorrect-report-routing [25] NULL
sf-incorrect-monitor-name [26] NULL
sf-context-not-available [27] NULL
sf-transmission-error [28] NULL

```

END

The “jtm-user-code” shall be the CHOICE “absent” unless this International Standard is referenced by another standard which specifies the abstract and possible transfer syntaxes for the embedded context “user-defined”

NOTES

- 1 Section 3 specifies the circumstances in which each of these error codes is generated. Those marked “sf” are service failures, “ue” are user errors, “ua” are user actions, “rl” are retry-later reasons and “w” are warnings.
- 2 Future addenda are likely to increase the range of codes that a Basic Class implementation can receive. Implementations should therefore be able to accept additional tagged NULL values. When an implementation is converting a JTM-code for human consumption, a code whose interpretation is unknown should be presented in such a way as to allow the tag used to be identified.

This datatype is embedded in the “CCR-diagnostics” datatype defined in 2.3.2.

2.3.2 CCR diagnostics

CCR diagnostics are carried in user data of CCR primitives (see clause 2.4), in JTM diagnostics in reports (see 2.5.7), and in embedded diagnostics in Transfer-elements (see 2.5.5).

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
CCR-diagnostics ::=
{warning
    {generator
        code
        reason
    }
no-retry
    {generator
        code
        reason
    }
retry-later
    {retry-timer
        {retry-reason
            {generator
                code
                reason
            }
        }
    }
}
CHOICE
[0] SET OF SET
    [0] JTM-name
    [1] JTM-code
    [2] Message
-- one or more in the SET OF --
[1] SET OF SET
    [0] JTM-name
    [1] JTM-code
    [2] Message
-- one or more in the SET OF --
[2] SET
    [0] INTEGER
        OPTIONAL
    [1] SET OF SET
        [0] JTM-name
        [1] JTM-code
        [2] Message
-- one or more in the SET OF --
END

```

2.4 User data on CCR primitives

These datatypes shall be the values used by JTM for the user data fields declared as "EXTERNAL" in ISO 9805.

ISO 8832 : 1989 (E)

2.4.1 User data on C-BEGIN request and indication

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
C-BEGIN-USER-DATA ::= SET
{
  commitment-level      [0] INTEGER ,
  diagnostic-code-indicator [1] SEQUENCE OF
    -- Zero, one or more in the SEQUENCE OF
    Code-specifier
}

Code-specifier ::=
CHOICE
{
  any-characters      [0] IMPLICIT NULL ,
  code-sets           [1] IMPLICIT SET OF
    -- one or more in the SET OF
    INTEGER
}

END

```

An implementation shall generate "Messages" (see 2.2.6) using only those graphics characters specified in any one of the "code-specifiers", or using only those graphics characters in the International Reference Version of ISO 646.

The CHOICE "any-characters" permits any character set to be used. The INTEGER in "code-sets" references the register entry with that number in the ISO Register of Character Sets, and permits all characters registered with that entry number.

Commitment-level is defined in ISO 8831 and shall take only the values AGENCY-ACCEPTANCE and PROVIDER-ACCEPTANCE. Section 3 specifies the procedures related to commitment-level.

2.4.2 User data on C-READY request and indication

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
C-READY-USER-DATA ::= SET
{
  commitment-level      [0] INTEGER ,
  warnings              [1]
    warning<CCR-diagnostics OPTIONAL ,
  accounting-info       [2] ANY OPTIONAL
}

END

```

"accounting-info", if present when a C-READY indication is issued to a Basic Class implementation, shall be ignored. It shall not be present when a C-READY request is issued by a Basic Class implementation.

NOTE — A future amendment to this International Standard is expected to permit its presence and processing provided its use and form conforms to the JTM extended class standard

The commitment-level field can take all values, including COMPLETION.

2.4.3 User data on C-REFUSE request and indication

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
C-REFUSE-USER-DATA ::= SET
{
  diagnostic      [0] CHOICE
    {
      retry-later<CCR-diagnostics ,
      no-retry<CCR-diagnostics
    },
  accounting-info [1] ANY OPTIONAL
}

END

```

The text of 2.4.2 shall apply to the "accounting-info" field.

2.4.4 User data on C-PREPARE request and indication

This primitive is not used by JTM.

2.4.5 User data on C-RESTART request and indication

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
C-RESTART-RI-USER-DATA ::= NULL
END

```

2.4.6 User data on C-RESTART response and confirm

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
C-RESTART-RC-USER-DATA ::= CHOICE
{
  done      [0] NULL ,
  refuse    [1] SET
    {
      diagnostic      [0] CHOICE
        {
          retry-later<CCR-diagnostics ,
          no-retry<CCR-diagnostics
        },
      accounting-info [1] ANY OPTIONAL ,
      retry-later     [2]
        retry-later<CCR-diagnostics ,
      action          [3] NULL
    }
}

END

```

The CHOICE shall be that with the ASN.1 identifier corresponding to the value of the CCR resumption point parameter.

The text of 2.4.2 shall apply to the "accounting-info" field.

2.5 Transfer elements

This clause defines a syntactic structure, and specifies semantic restrictions for a "Transfer-element". The identification of fields of this datatype are used to identify corresponding fields in the work specification.

A JTM transfer, as specified in section 3, consists of a single P-DATA primitive (within a CCR atomic action), carrying in its user data parameter the following presentation data values:

- a value of the "Transfer-element" datatype; optionally followed by
- the presentation data values specified in a document type definition to carry the semantics of a single document.

NOTE — In JTM Basic Class each work specification contains at most one document, so questions of how the end of one document is distinguished from the start of the next do not arise

2.5.1 Top-level fields

2.5.1.1 Syntactic definitions

ISO8832-JTM DEFINITIONS ::=

BEGIN

Transfer-element ::= [APPLICATION 0] SEQUENCE

```
{osi-job-submission-system [0] JTM-name ,
  initiating-identification [1] Identification ,
  initiating-time [2] Time-stamp ,
  osi-job-name [3] GraphicString ,
  osi-job-local-reference [4] GraphicString ,
  audit-trace [5] SEQUENCE OF
    Audit-element ,
  primary-monitoring-specification
    [6]
    Monitoring-specification ,
  authorisations [7] SET OF
    Authorisation-element ,
  permissions [8] SET OF
    Permission-element ,
  subjob-name-list [9] Subjob-name-list ,
    COMPONENTS OF
    Subjob-specification ,
  proforma-list [10] SEQUENCE OF
    Proforma }
```

Subjob-name-list ::= SEQUENCE OF SEQUENCE

```
{proforma-name GraphicString ,
  qualifying-integer INTEGER }
```

Subjob-specification ::= SEQUENCE

```
{target [20] JTM-name ,
  type [21] Subjob-type ,
  urgency [22] CHOICE
    {medium [1] NULL },
  error-action [23] CHOICE
    {terminate [0] NULL },
  actions [24]
    JTM-action-parameters }
```

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 ,
  proforma-body [1] Proforma-specification }
```

Proforma-specification ::= SEQUENCE

```
{
    COMPONENTS OF
    Subjob-specification ,
  proforma-list [1] ANY }
```

END

The following data-types are defined in later sub-clauses:

Identification	2.5.4
Time-stamp	2.5.5
Audit-element	2.5.2
Monitoring-specification	2.5.3
Authorisation-element	2.5.4
Permission-element	2.5.4
Document-movement-operations	2.5.5
Work-manipulation-operations	2.5.6
Report-movement-operation	2.5.7

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 indicator".

Finally, it can (but might not) contain a document which is transferred as an additional value on the P-DATA primitive.

When a work specification is awaiting a J-END-SIGNAL, the outermost "Subjob-specification" is null.

2.5.1.2 Semantic restrictions

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

a) the value of the "osi-job-submission-system JTM-name" shall equal the value of the first "JTM-name" in the "audit-trace" (see 2.5.2);

b) the "qualifying-integer" 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" and shall not be "work-manipulation";

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 2.5.5);

ISO 8832 : 1989 (E)

- i) the P-DATA shall contain at most one value (document) following the Transfer-element;

2.5.2 Audit elements

2.5.2.1 Syntactic definitions

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Audit-element ::=
SEQUENCE
{ sender [0] JTM-name ,
  status [1] CHOICE
    { unknown [0] NULL ,
      known [1] NULL ,
      authenticated [2] NULL } }
END
```

2.5.2.2 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 3.

2.5.3 Monitoring specifications

2.5.3.1 Syntactic definitions

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Monitoring-specification ::= SEQUENCE
{ monitor-specification [0] Monitor-spec ,
  report-selector [1] BIT STRING
    { normal-termination (0) ,
      manipulation-termination (1) ,
      abnormal-termination (2) ,
      user-message (3) } }

Monitor-spec ::= SEQUENCE
{ monitor-system-name [0] JTM-name ,
  disposal-instructions CHOICE
    { disposal-data [1] SEQUENCE
      { agency [0] Se-identification ,
        doc-name [1] Document-se-reference } ,
      keep [2] NULL } }
END
```

The following data-types are defined in later sub-clauses:

Se-identification	2.5.5
Document-se-reference	2.5.5

2.5.3.2 Semantic restrictions

The "report-selector" shall be a bit-string of a length chosen by the sender. A bit in the bit-string shall not be set to 1 unless it is one of the distinguished bits named in the ASN.1 definition.

If the "monitor-system-name" is a Basic Class implementation, the "disposal-instructions" shall be "disposal-data".

NOTE — The value "keep NULL" is only present in Basic Class transfer-elements that originated at extended implementations.

2.5.4 Authorisation and permission elements

2.5.4.1 Syntactic definitions

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Authorisation-element ::= SEQUENCE
{ id [0] Identification ,
  validation [1] CHOICE
    { checked-index [0] INTEGER ,
      password [1] CHOICE
        { unset [0] NULL ,
          graphics [1] GraphicString ,
          binary [2] OCTET STRING } } }

Permission-element ::= Identification

Identification ::= CHOICE
{ open-system [0] JTM-name ,
  authority [1] User-identification-authority ,
  user [2] SEQUENCE
    { authority [0] User-identification-authority ,
      id [1] User-identification } }
END
```

2.5.4.2 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".

2.5.5 Document movement operations

2.5.5.1 Syntactic definitions

```
ISO8832-JTM DEFINITIONS ::=
BEGIN
Document-movement-operations ::= SEQUENCE OF
  Document-movement

Document-movement ::= SEQUENCE
{ type [0] OBJECT IDENTIFIER ,
  ses [1] SEQUENCE OF
    Se-identification }

document-block ::= CHOICE
{ [2] Single-form }

Se-identification ::= CHOICE
{ jtm-se [0] SEQUENCE
  { se-name [0] Agency-name } }

Single-form ::= SEQUENCE
{ doc-name [0] Document-se-reference ,
  docs [1] SEQUENCE OF
    Document-pointer }

Document-se-reference ::= CHOICE
{ jtm-write-data [0] SEQUENCE
  { document-name [0] Name-list ,
    se-access-parameter [1] CHOICE
      { normal [0] NULL ,
        add [1] NULL } } }
```

```

Document-pointer ::= CHOICE
{ embedded [0] NULL ,
  single-document-reference [1] SEQUENCE
    { action-open-system [0] JTM-name ,
      source [1] Source-identification ,
      doc-name [2]
        Document-source-reference ,
      embedded-diagnostics [3] CHOICE
        { embed [0] NULL ,
          state [4] CHOICE
            { not-attempted [0] NULL ,
              failed [1] SEQUENCE
                { time [0] Time-stamp ,
                  diagnostics [1]
                    no-retry<CCR-diagnostics > } } } } } }

```

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 ,
  jtm-source [1] SEQUENCE
    { source-name [0] Agency-name } }

```

```

Document-source-reference ::= CHOICE
{ jtm-read-data [0] SEQUENCE
  { document-name [0] Name-list ,
    source-access-parameter [1] CHOICE
      { move [0] NULL } } }

```

```

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

```

2.5.5.2 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. (There is at most one such occurrence in the Basic Class.)

The type of the document shall agree with the “Document-type” given in the “Document-movement” containing the “embedded” CHOICE of “Document-pointer”.

The “Document-movement-operations” shall consist of exactly one “document-movement” and the “ses” shall consist of exactly one “Se-identification”.

If a “Document-se-reference” is in a “Monitor-spec”, the “se-access-parameter” shall be “add”. If a “Document-se-reference” is in a “single-form”, the “se-access-parameter” shall be “normal”.

The “SEQUENCE OF Document-pointer” in the “Single-form” shall be a single “Document-pointer”.

In an outer level subjob specification of a work specification whose target is this JTM ASE, the single document movement shall either have the “state” “failed”, or shall have the following generic value:

```

{ type t ,
  ses
    { jtm-se n } ,
  document-block
    { doc-name d ,
      { document-name d
        { embedded NULL } } } }

```

where *t* is an instance of “Document-type” set to

```

{ iso jtm simple text document }
or { iso jtm simple print document }
or { iso jtm work-display document }

```

n 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”.

2.5.5.2.1 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” and for which the target is this JTM ASE shall have the value

```

{ type jtm work-display NULL ,
  ses
    { jtm-se n } ,
  document-block
    { doc-name jtm-write-data d ,
      docs { single-document-reference
        { action-open-system w ,
          source { provider NULL } ,
          doc-name jtm-read-data
            { document name { s } ,
              state not-attempted NULL } } } } }

```

n is any “GraphicString”, *d* is any “Name-list”, *w* is equal to the “JTM-name” of the top-level “target” in the “Transfer-element” and *s* is a “GraphicString” with the value “DISPLAY”.

2.5.5.2.2 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.

2.5.6 Work-manipulation operations

2.5.6.1 Syntactic definitions

```

ISO8832-JTM DEFINITIONS ::=
BEGIN

```

```

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

```

ISO 8832 : 1989 (E)

```

Work-operation ::= CHOICE
{ select      [0] Selector      ,
  kill        [1] NULL          ,
  stop        [2] NULL          ,
  display     CHOICE            ,
  { brief     [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 } }

Field-test ::= CHOICE
{ osi-job-submission-system-equals
  [0] JTM-name ,
  osi-jobname-equals [1] GraphicString ,
  subjob-type-equals [2] Subjob-type }
END

```

2.5.6.2 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 a "GraphicString" with the value "DISPLAY", 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 and-clause
  { header-one osi-job-name-equals q ,
    header-two subjob-type-equals r } } }

```

where *p* is any instance of an "JTM-name", *q* is any instance of a "GraphicString", and *r* is any instance of a "Subjob-type".

2.5.7 Report-movement operation

2.5.7.1 Syntactic definitions

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
Report-movement-operation ::= SET OF SEQUENCE
{ monitor-indices [0] SET OF INTEGER ,
  report          [1] Single-report   }

```

```

Single-report ::= SEQUENCE
{ name-of-reporter [0] JTM-name ,
  time             [1] Time-stamp ,
  osi-job-submission-system [2] JTM-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-identification [9] CHOICE
  { normal-termination [0] SEQUENCE
    { text [0] Message ,
      spawn-count [1] INTEGER } ,
  manipulation-termination [1] SEQUENCE
  { text [0] Message ,
    spawn-count [1] INTEGER } ,
  abnormal-termination [2] SEQUENCE
  { text [0] Message ,
    spawn-count [1] INTEGER ,
    errors [2] SET OF
      Diagnostic-information } ,
  user-message [3] SEQUENCE
  { text [0] Message } } }

```

Diagnostic-information ::= SEQUENCE

```

{ generator details [0] CHOICE
  { source-details [0] SEQUENCE
    { agency [0] Source-identification ,
      doc-name [1]
        Document-source-reference } ,
  se-details [1] SEQUENCE
  { agency [0] Se-identification ,
    doc-name [1]
      Document-se-reference } ,
  recipient-details [2] JTM-name ,
  provider [3] NULL ,
  when [1] Time-stamp ,
  diagnostics [2]
    no-retry<CCR-diagnostics }

```

END

2.5.7.2 Semantic restrictions

The "monitor-indices" shall consists of the single integer zero.

The "Subjob-name-list" shall satisfy the restrictions of 2.5.1.2(b).

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

2.6 Work-display and report-display documents

2.6.1 Syntactic definitions

ISO8832-JTM DEFINITIONS ::=

BEGIN

```

Work-display-doc ::= [APPLICATION 4] SEQUENCE OF
  SEQUENCE
  { displaying-system [0] JTM-name ,
    time [1] Time-stamp ,
    display [2] SEQUENCE OF
      CHOICE
      { [0] Brief-display } }

```

```

Brief-display ::=
{ details
  { osi-job-submission-system
    initiating-identification
    initiating-time
    osi-job-name
    osi-job-local-reference
    subjob-name-list
    type
    status
  }
}

SEQUENCE
[0] SEQUENCE
[0] JTM-name
[1] Identification
[2] Time-stamp
[3] GraphicString
[4] GraphicString
[5] Subjob-name-list
[6] Subjob-type
[1] SET OF
Destination-status

Destination-status ::=
{ destination
  { recipient
    source-agency
    { agency
      se-agency
      { agency
        status
        { in-progress
          accepted
          waiting
          {
            retry-later<CCR-diagnostics OPTIONAL }
            time
          }
        }
      }
    }
  }
}

SEQUENCE
[0] CHOICE
[0] JTM-name
[1] CHOICE
[0] Agency-name
[2] CHOICE
[0] Agency-name
[1] CHOICE
[0] Message
[1] Message
[2] SEQUENCE
[0] Message
[1]
[2] Time-stamp

Report-display-doc ::= [APPLICATION 5] SEQUENCE OF
Report-display

Report-display ::=
{ monitor-system
  time-of-display
}

SEQUENCE
[0] JTM-name
[1] Time-stamp
[2] SEQUENCE OF
Single-report

END

```

2.6.2 Semantic restrictions

A "status" of "accepted" shall not occur unless the "destination" is "se-agency".

2.7 Summary of datatypes

In this clause the following datatypes are declared as ANY (see the text of earlier clauses for further discussion):

JTM-name
User-identification-authority

In this summary, a new line is not significant, and a semi-colon is used to terminate a datatype definition. The order of datatype definitions is not significant. Where conflicts arise between this clause and earlier clauses, the earlier clauses take precedence.

```

ISO8832-JTM DEFINITIONS ::=
BEGIN
JTM-name ::= Application-entity-title;
User-identification-authority ::= Application-entity-title;
Application-entity-title ::=
  ISO8650-ACSE-1.ApplicationTitle;
Agency-name ::= GraphicString;
User-identification ::= GraphicString;
Name-list ::= SEQUENCE OF GraphicString;

```

Message ::= SEQUENCE OF GraphicString;

```

JTM-code ::=
{ jtm-user-code
  { absent
    user-defined
  }
}

SEQUENCE
CHOICE
[0] NULL
[1] EXTERNAL
CHOICE
[0] NULL
[1] NULL
[2] NULL
[3] NULL
[4] NULL
[5] NULL
[6] NULL
[7] NULL
[8] NULL
[9] NULL
[10] NULL
[11] NULL
[12] NULL
[13] NULL
[14] NULL
[15] NULL
[16] NULL
[17] NULL
[18] NULL
[19] NULL
[20] NULL
[21] NULL
[22] NULL
[23] NULL
[24] NULL
[25] NULL
[26] NULL
[27] NULL
[28] NULL
};

```

```

CCR-diagnostics ::=
{ warning
  { generator
    code
    reason
  }
}

CHOICE
[0] SET OF SET
[0] JTM-name
[1] JTM-code
[2] Message
-- one or more in the SET OF --
[1] SET OF SET
[0] JTM-name
[1] JTM-code
[2] Message
-- one or more in the SET OF --
[2] SET
[0] INTEGER
OPTIONAL
[1] SET OF SET
[0] JTM-name
[1] JTM-code
[2] Message
-- one or more in the SET OF --
};

```

```

C-BEGIN-USER-DATA ::= SET
{ commitment-level
  diagnostic-code-indicator
  -- Zero, one or more in the SEQUENCE OF
    Code-specifier
  }
};

```

```

Code-specifier ::=
{ any-characters
  code-sets
  -- one or more in the SET OF
    INTEGER
  }
};

```

```

C-READY-USER-DATA ::= SET
{ commitment-level
  warnings
  [0] INTEGER
  [1]

```

ISO 8832 : 1989 (E)

warning<CCR-diagnostics OPTIONAL ,
 accounting-info [2] ANY OPTIONAL } ;
 C-REFUSE-USER-DATA ::= SET
 { diagnostic [0] CHOICE
 { retry-later<CCR-diagnostics ,
 no-retry<CCR-diagnostics } ,
 accounting-info [1] ANY OPTIONAL } ;
 C-RESTART-RI-USER-DATA ::= NULL ;
 C-RESTART-RC-USER-DATA ::= CHOICE
 { done [0] NULL ,
 refuse [1] SET
 { diagnostic [0] CHOICE
 { retry-later<CCR-diagnostics ,
 no-retry<CCR-diagnostics } ,
 accounting-info [1] ANY OPTIONAL } ,
 retry-later [2]
 retry-later<CCR-diagnostics } ,
 action [3] NULL } ;
 Transfer-element ::= [APPLICATION 0] SEQUENCE
 { osi-job-submission-system [0] JTM-name ,
 initiating-identification [1] Identification ,
 initiating-time [2] Time-stamp ,
 osi-job-name [3] GraphicString ,
 osi-job-local-reference [4] GraphicString ,
 audit-trace [5] SEQUENCE OF
 Audit-element ,
 primary-monitoring-specification [6]
 Monitoring-specification ,
 authorisations [7] SET OF
 Authorisation-element ,
 permissions [8] SET OF
 Permission-element ,
 subjob-name-list [9] Subjob-name-list ,
 COMPONENTS OF
 Subjob-specification ,
 proforma-list [10] SEQUENCE OF
 Proforma } ;
 Subjob-name-list ::= SEQUENCE OF SEQUENCE
 { proforma-name GraphicString ,
 qualifying-integer INTEGER } ;
 Subjob-specification ::= SEQUENCE
 { target [20] JTM-name ,
 type [21] Subjob-type ,
 urgency [22] CHOICE
 { medium [1] NULL } ,
 error-action [23] CHOICE
 { terminate [0] NULL } ,
 actions [24]
 JTM-action-parameters } ;
 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 ,
 proforma-body [1] Proforma-specification } ;
 Proforma-specification ::= SEQUENCE
 { COMPONENTS OF
 Subjob-specification ,
 proforma-list [1] ANY } ;
 Audit-element ::= SEQUENCE

{ sender [0] JTM-name ,
 status [1] CHOICE
 { unknown [0] NULL ,
 known [1] NULL ,
 authenticated [2] NULL } } ;
 Monitoring-specification ::= SEQUENCE
 { monitor-specification [0] Monitor-spec ,
 report-selector [1] BIT STRING
 { normal-termination (0) ,
 manipulation-termination (1) ,
 abnormal-termination (2) ,
 user-message (3) } } ;
 Monitor-spec ::= SEQUENCE
 { monitor-system-name [0] JTM-name ,
 disposal-instructions CHOICE
 { disposal-data [1] SEQUENCE
 { agency [0] Se-identification ,
 doc-name [1]
 Document-se-reference } ,
 keep [2] NULL } } ;
 Authorisation-element ::= SEQUENCE
 { id [0] Identification ,
 validation [1] CHOICE
 { checked-index [0] INTEGER ,
 password [1] CHOICE
 { unset [0] NULL ,
 graphics [1] GraphicString ,
 binary [2]
 OCTET STRING } } } ;
 Permission-element ::= Identification ;
 Identification ::= CHOICE
 { [0] JTM-name ,
 [1]
 User-identification-authority ,
 user [2] SEQUENCE
 { authority [0]
 User-identification-authority ,
 id [1] User-identification } } ;
 Document-movement-operations ::= SEQUENCE OF
 Document-movement ;
 Document-movement ::= SEQUENCE
 { type [0] OBJECT IDENTIFIER ,
 ses [1] SEQUENCE OF
 Se-identification ,
 document-block CHOICE
 { [2] Single-form } } ;
 Se-identification ::= CHOICE
 { jtm-se [0] SEQUENCE
 { se-name [0] Agency-name } } ;
 Single-form ::= SEQUENCE
 { doc-name [0] Document-se-reference ,
 docs [1] SEQUENCE OF
 Document-pointer } ;
 Document-se-reference ::= CHOICE
 { jtm-write-data [0] SEQUENCE
 { document-name [0] Name-list ,
 se-access-parameter [1] CHOICE
 { normal [0] NULL ,
 add [1] NULL } } } ;
 Document-pointer ::= CHOICE
 { embedded [0] NULL ,
 single-document-reference [1] SEQUENCE
 { action-open-system [0] JTM-name ,
 source [1] Source-identification ,
 doc-name [2]
 Document-source-reference ,
 embedded-diagnostics [3] CHOICE
 { embed [0] NULL ,
 state [4] CHOICE