

Implementation of
ISO/IEC 8832:1992

Information technology — Open Systems Interconnection — Specification of the Basic Class and Full Protocol for job transfer and manipulation

UDC 681.3:621.39

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW



Document Sharing Hub

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/21, upon which the following bodies were represented:

- AEA Technology
- British Computer Society
- British Telecommunications plc
- Department of Trade and Industry (IT Standards Unit)
- Digital Equipment Co. Ltd.
- HM Treasury (Central Computer and Telecommunications Agency)
- IBM United Kingdom Ltd.
- Information Systems Committee of the Universities Funding Council
- International Computers Limited
- Level 7 Ltd.
- Logica UK Ltd.
- Ministry of Defence
- National Health Services
- PSC International Ltd.
- Rank Xerox Ltd.
- SEMA Group Systems Ltd.
- Salford Software services (3-5)
- Swift Project Consortium

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Standards Board and comes into effect on 15 June 1992

© BSI 03-2000

First published October 1990
Second edition June 1992

The following BSI references relate to the work on this standard:
Committee reference IST/21
Draft for comment 90/67413 DC

ISBN 0 580 20923 7

Amendments issued since publication

Amd. No.	Date	Comments

Contents

	Page
Committees responsible	Inside front cover
National foreword	ii
Foreword	vi
Text of ISO/IEC 8832	1

National foreword

This British Standard reproduces verbatim ISO/IEC 8832:1992 and implements it as the UK national standard. It supersedes BS ISO 8832:1989 which is withdrawn.

This British Standard is published under the direction of the Information Systems Technology Standards Policy Committee whose Technical Committee IST/21 has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international committee any enquiries on interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

NOTE International and European Standards, as well as overseas standards, are available from BSI Sales Department, BSI, Linford Wood, Milton Keynes, MK14 6LE.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the ISO/IEC title page, pages ii to vi, pages 1 to 149 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

INTERNATIONAL STANDARD

ISO/IEC
8832

Second edition
1992-03-01

Information technology — Open Systems Interconnection — Specification of the Basic Class and Full Protocol for job transfer and manipulation

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



Reference number
ISO/IEC 8832:1992(E)

Contents

	Page
Foreword	vi
Introduction	1
Section 1. General	
1.1 Scope	3
1.2 Normative references	3
1.3 Definitions	4
1.4 Abbreviations	5
1.5 Relation of JTM to other services	5
1.5.1 JTM architecture	5
1.5.2 JTM ASEs and agencies	6
1.5.3 Use of the Presentation Service	6
1.5.4 Use of the ACSE ASE	6
1.5.5 Use of the CCR ASE	6
1.5.6 Service primitives referenced by this International Standard	7
1.5.7 Summary of JTM architecture	7
1.6 Conformance	8
Section 2. JTM datatypes	
2.1 Introduction to JTM datatype definitions	9
2.2 Names and messages	9
2.2.1 Global names	9
2.2.2 Names local to a JTM ASE	10
2.2.3 Names local to a User-identification-authority	10
2.2.4 Name lists	10
2.2.5 Context names	10
2.2.6 Human-readable messages	12
2.3 Document separators	12
2.4 Diagnostics	12
2.4.1 JTM Diagnostic codes	12
2.4.2 CCR diagnostics	13
2.5 User data on CCR primitives	14
2.5.1 User data on C-BEGIN request and indication	14
2.5.2 User data on C-READY request and indication	15
2.5.3 User data on C-ROLLBACK request and indication	15
2.6 Transfer elements	15
2.6.1 Top-level fields	16
2.6.2 Audit elements	19
2.6.3 Monitoring specifications	19
2.6.4 Authorisation, permission, account and hold elements	20
2.6.5 Document movement operations	21
2.6.6 Work-manipulation operations	25
2.6.7 Report-movement operation	29
2.6.8 Transfer-manipulation operations	30
2.6.9 Report-manipulation-operations	31
2.7 Work-display, report-display and tcr-display documents	31
2.7.1 Syntactic definitions	31
2.7.2 Semantic restrictions	33
2.8 Summary of datatypes	34
Section 3. JTM procedures	
3.1 Introduction to the procedures	50
3.1.1 General requirements	50

	Page
3.2 Processing of J-INITIATE requests	50
3.2.1 Creation of work specifications	51
3.2.2 Authorisations	52
3.2.3 Permissions	52
3.2.4 Accounts	52
3.2.5 Security parameters	53
3.2.6 Holds	53
3.2.7 JTM-action-parameters	53
3.2.8 Proformas	53
3.2.9 Creation report	53
3.2.10 Accounting report	53
3.2.11 J-INITIATE confirm	53
3.3 Procedures for receipt of work specifications	53
3.4 Initial processing of a work specification	56
3.5 Deferred processing of a work specification	58
3.6 Reference resolution	61
3.7 Procedures for transmission of work specifications	65
3.8 Document movement procedures	68
3.9 Report movement procedures	71
3.10 Work manipulation procedures	72
3.11 Report-manipulation operations	84
3.12 Transfer-control manipulation operations	86
3.13 Action on J-END-SIGNAL requests	88
3.14 Spawning procedures	89
3.15 Procedure for report generation as a separate atomic action	92
3.16 Procedure for report generation as part of same atomic action	93
3.17 Action on J-MESSAGE requests	95
3.18 Action on J-SPAWN requests	96
3.19 Holding and releasing work specifications	96
3.20 Provider initiation of tcr-setting	97
3.21 Provider initiation of tcr-check	99
Section 4. JTM functionality	
4.1 Static conformance requirements	101
4.2 Functional classes	102
4.2.1 Support for Basic Class OSI-job submission	103
4.2.2 Full support for JTM work OSI-job submission	103
4.2.3 Support for JTM monitoring	104
4.2.4 Support for JTM monitor point storage	104
4.2.5 Support for Basic Class JTM manipulation	104
4.2.6 Support for JTM work-manipulation processing (top-level only)	104
4.2.7 Full support for JTM work-manipulation processing	104
4.2.8 Support for report-manipulation	104
4.2.9 Support for transfer-control manipulation	105
4.2.10 Full support for JTM user OSI-job submission	105
4.2.11 Basic Class JTM support for a local file-store	105
4.2.12 JTM support for a local file-store (single file)	106
4.2.13 JTM support for a local file-store (transparent storage)	106
4.2.14 JTM support for a local file-store (multiple files)	106

	Page
4.2.15 Full JTM support for a local-filestore	107
4.2.16 Basic Class JTM support for an output device	107
4.2.17 Full JTM support for an output device	107
4.2.18 Basic Class JTM support for job processing	107
4.2.19 Full JTM support for job processing	107
4.2.20 Basic class JTM support from language xyz	108
4.2.21 Full JTM support from language xyz	108
4.2.22 Support for JTM transfer-control, without tcr-checking	108
4.2.23 Full support for JTM transfer-control	108
4.2.24 Support for JTM transfer manipulation	108
4.2.25 Support for a JTM process	108
4.2.26 Support for a store-and-forward relay	109
4.2.27 Complete Basic Class JTM Support	109
4.3 Management function requirements	109
4.4 Documentation	109
Section 5. JTM transfers	
5.1 Application association control	111
5.1.1 Notation	111
5.1.2 Procedures	112
5.1.3 ASSOcnf-return codes	115
5.2 Parameters in Association Control Service Primitives	115
5.2.1 Parameters in A-ASSOCIATE service primitives	115
5.2.2 Parameters in A-RELEASE service primitives	118
5.3 P-service parameters	118
5.3.1 Parameter of P-DATA	119
5.3.2 Interruption of P-DATA	119
5.4 C-service parameters	119
5.4.1 C-BEGIN and J-BEGIN request and indication	119
5.4.2 C-READY and J-READY request and indication	120
5.4.3 C-ROLLBACK and J-ROLLBACK request and indication	120
5.4.4 C-RECOVER and J-RECOVER	121
Annex A (normative) Management functions	122
Annex B (normative) Document Types	125
B.1 Simple text document type	125
B.2 Simple print document type	126
B.3 Simple binary document type	128
B.4 Work-display document type	130
B.5 Report-display document type	131
B.6 Transfer-display document type	132
Annex C (informative) JTM Testing Procedures	133
Annex D (informative) Summary of ASN.1 OBJECT IDENTIFIER assignments	137
Annex E (informative) Tutorial examples of protocol sequences	137
Annex F (informative) Basic Class datatypes	142
Figure 1 — JTM upper layer architecture	8
Figure C.1 — JTM testing architecture	135
Figure E.1 — An example of Job Transfer	138

	Page
Figure E.2 — Service primitive sequence for Figure E.1 using provider acceptance	139
Figure E.3 — Time sequence diagram for the atomic actions shown in Figure E.2	140
Figure E.4 — Service primitive sequence for Figure E.1 using agency acceptance	141
Figure E.5 — Time sequence diagram for the atomic actions shown in Figure E.4	142
Table 1 — Se-access-parameter and file existence	106
Table 2 — Incoming events	112
Table 3 — States of the procedures	112
Table 4 — Predicates	113
Table 5 — Outgoing events	113
Table 6 — State/event table	114
Table 7 — Parameter values in A-ASSOCIATE request/indication	116
Table 8 — Parameter values in A-ASSOCIATE response/confirm	118
Table 9 — A-RELEASE request/indication parameters	119
Table 10 — A-RELEASE response/confirm parameters	119

Descriptors: Data processing, information interchange, network interconnection, open systems interconnection, communication procedure.

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8832 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO 8832:1989), which has been technically revised.

Annex A and Annex B form an integral part of this International Standard.

Annex C, Annex D, Annex E and Annex F are for information only.

Introduction

This International Standard specifies the behaviour of an Application Service Element which supports the Job Transfer and Manipulation (JTM) Service defined in ISO/IEC 8831, for both the Basic Class and full JTM services.

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/IEC 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**. The application-context for the JTM Basic Class procedures is called "ISO JTM Basic Class Application-context", and that for the full JTM procedures is called "ISO JTM Full Application-context". One of these is established, prior to the transfer of a JTM Transfer Element, using the services of ISO 8649.

When two implementations of either JTM 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 4 of this International Standard, and a mandatory set for the CCR presentation-context are specified in ISO/IEC 9805. The mandatory set for documents are specified with the definition of the document type in Annex B.

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

NOTE The JTM International Standards were developed before the concept of a Protocol Implementation Conformance Statement (PICS). The documentation requirements of section 4 are broadly equivalent.

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 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 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.

Annex F is not part of this International Standard and presents the ASN.1 datatypes that are used in Basic Class JTM as if they were defined independently of the full datatypes.

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 invoked 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 2375:1985, *Data processing — Procedures for the registration of escape sequences.*

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

ISO 8571-4:1988, *Information processing systems — Open systems interconnection — File Transfer, Access and Management — Part 4: File Protocol Specification.*

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/IEC 8824:1990, *Information technology — Open Systems Interconnection — Specification of abstract syntax notation one (ASN.1).*

ISO/IEC 8825:1990, *Information technology — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).*

ISO/IEC 8831:1992, *Information technology — Open Systems Interconnection — Job transfer and manipulation concepts and services.*

ISO/IEC 9804:1990, *Information technology — Open Systems Interconnection — Service definition for the Commitment, Concurrency and Recovery service element.*

ISO/IEC 9805:1990, *Information technology — 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/IEC 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 JTM Application Service Element.

This ASE can operate as part of an application-entity which contains only the 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.

The 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 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 JTM ASE provides services directly to the user element. Where the 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.

In Basic Class, the entire transfer element, together with any associated documents, is transferred using a single P-DATA primitive. In full JTM, an indefinite number of P-DATA primitives can be used. The P-DATA primitives are 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 either “ISO JTM Basic Class Application-context” or “ISO JTM Full Application-context”.

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

When an extended JTM implementation initiates an application association, it may propose use of “ISO JTM Full Application-context”. If the responding implementation is Basic Class, the negotiation specified in section 5 will result in the establishment of “ISO JTM Basic Class Application-context”. If the responder is an extended JTM implementation, “ISO JTM Full Application-context” will be established.

Local scheduling requirements may cause abandonment of the JTM application-context (or of the underlying presentation connection). Later recovery (resynchronisation) is performed by use of C-RECOVER within a new “ISO JTM Basic Class Application-context” or “ISO JTM Full Application-context”, as appropriate 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.

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.

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” or “ISO JTM Full 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-INITIATE-REPORT-MAN request;
- e) a J-INITIATE-TCR-MAN request;
- f) a J-END-SIGNAL request;
- g) a J-MESSAGE request;
- h) J-SPAWN request;
- i) all CCR indications and confirms in the “ISO JTM Basic Class Application-context” or “ISO JTM Full Application-context”;
- j) 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

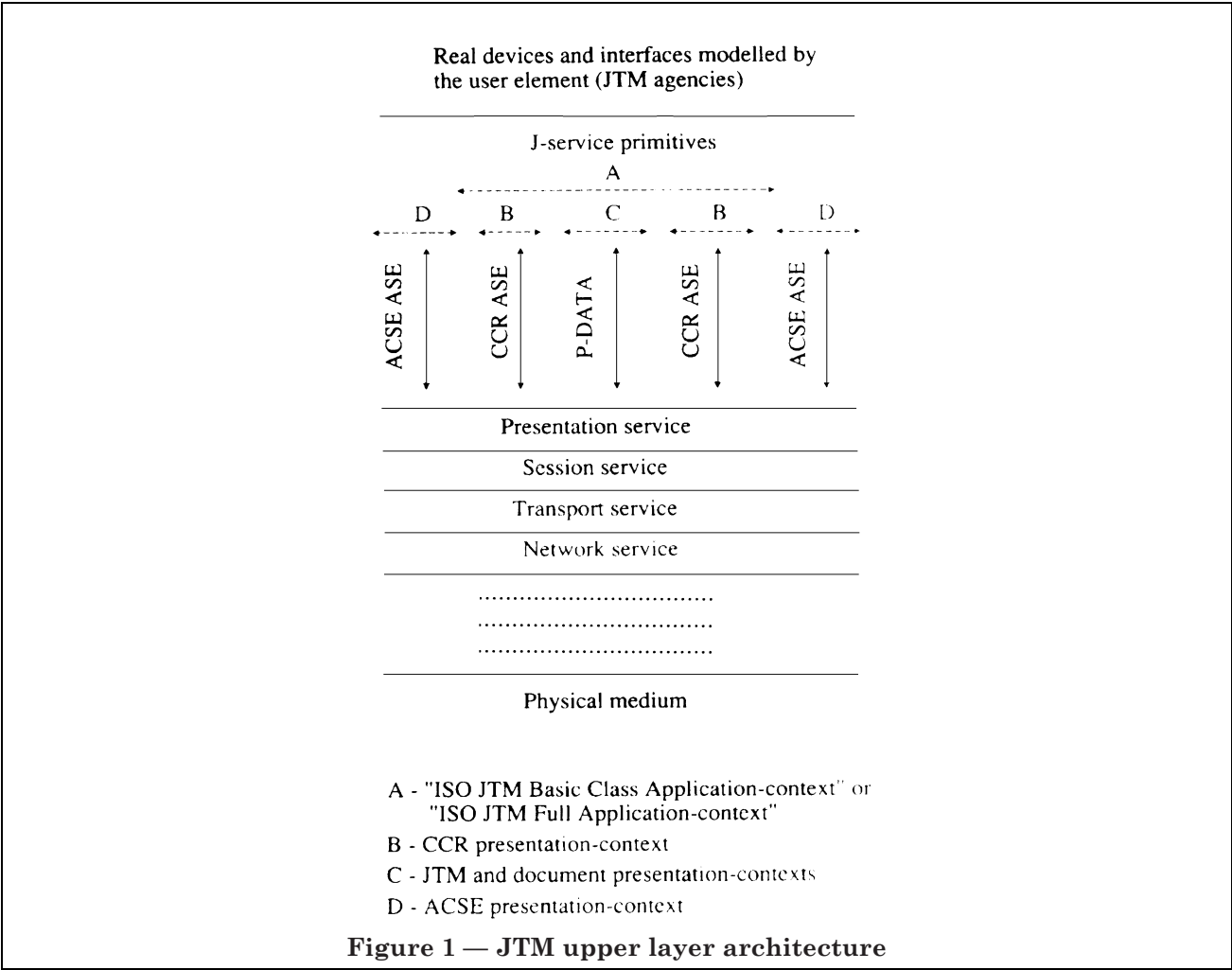
- a) a P-DATA request in “ISO JTM Basic Class Application-context” or “ISO JTM Full Application-context”;
- b) a J-DISPOSE indication and response;
- c) a J-GIVE indication and response;
- d) a J-ENQUIRE indication and response;
- e) a J-STATUS indication;
- f) a J-KILL indication;
- g) a J-STOP indication;
- h) a J-HOLD indication;
- i) a J-RELEASE indication;
- j) all CCR requests and responses;
- k) 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” and “ISO JTM Full 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” or “ISO JTM Full 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.

Section 2. JTM datatypes

2.1 Introduction to JTM datatype definitions

This section defines the JTM datatypes. It uses the standard notation of ISO/IEC 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 either a single document, or a number of documents and document separators. The abstract syntax of the document separators is defined in this section. The abstract syntax of the documents is defined elsewhere.

This section 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 datatype for the document separator.

Clause 2.4 defines the datatypes used for diagnostics and for accounting information.

Clause 2.5 defines the datatypes used for the user data parameters of CCR primitives.

Clause 2.6 defines a datatype called "Transfer-element". The JTM service is provided by generating, transferring and processing work specifications. A work specification (a semantic concept) is represented during transfer as a "Transfer-element" together with zero, one or more documents and document separators. The representation of a Basic Class work specification will not include more than one document and will include no document separators. Clause 2.6 defines the complete "Transfer-element" data structure for all CHOICES which are used in this International Standard, 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 2.7 defines the datatypes "Work-display-doc", "Report-display-doc" and "Tcr-display-doc". These are generated by the JTM service provider as specified in section 3, and are the only JTM-defined documents.

Clause 2.8 is a repetition of the datatype definitions in clauses 2.2, 2.3, 2.4, 2.5, 2.6 and 2.7, and is intended for machine processing. If differences occur, clauses 2.2, 2.3, 2.4, 2.5, 2.6, and 2.7 are authoritative.

2.2 Names and messages

2.2.1 Global names

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=

BEGIN

JTM-name ::= Application-entity-title

FTAM-name ::= Application-entity-title

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

Application-entity-title ::= ACSE-1.AE-title

END

NOTE 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.

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

NOTE 3 The datatype "FTAM-name" is not used in Basic Class.

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 IMPLICIT TAGS :=  
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 IMPLICIT TAGS :=  
BEGIN  
  User-identification ::= GraphicString  
END
```

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 IMPLICIT TAGS :=  
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 subclause 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 2.8; and
- c) the name of the transfer syntax for these datatypes which is obtained by applying the Basic Encoding Rules (see ISO/IEC 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 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 JTM implementation is aware of two application-context-names. The first identifies the procedures of this International Standard when the Basic Class restrictions are applied 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 of this International Standard 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 JTM systems (other than those specified by the ACSE and CCR ASEs) are either:

- a) values of the ASN.1 datatype “Transfer-element” defined in clause 2.6; or
- b) values of the ASN.1 datatypes
 - C-BEGIN-USER-DATA
 - C-READY-USER-DATA
 - C-ROLLBACK-RI-USER-DATA
 defined in clause 2.5; or
- c) presentation data values specified in a document type definition; or
- d) values of the ASN.1 datatype “Document-separator” defined in clause 2.3.

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), b) and d) 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”, “report-display” and “tcr-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/IEC 8825 (the Basic ASN.1 Encoding Rules) to the datatypes defined in this section:

{joint-iso-ccitt asn 1 (1) basic-encoding(1)}

“Basic encoding of a single ASN.1 type”

These ASN.1 object identifier and object descriptor values are assigned in ISO/IEC 8825. The following ASN.1 definitions also identify the same transfer-syntax, but their use is deprecated:

{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, 1.3.10.)

2.2.6 Human-readable messages

Messages are used to return diagnostic information to human users.

```
ISO 8832-JTM DEFINITIONS IMPLICIT TAGS :=
BEGIN
Message := SEQUENCE SIZE (1..MAX) OF GraphicString (SIZE (0..40))
END
```

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 **2.5**.)

NOTE 1 The requirement to support ISO 646 is subject to variation in national standards which are otherwise equivalent to this International Standard.

NOTE 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 Document separators

Document separators are used when a work specification contains more than one document in addition to the “Transfer-element”

```
ISO8832-JTM DEFINITIONS IMPLICIT TAGS :=
BEGIN
Document-separator := [APPLICATION 1] IMPLICIT NULL
END
```

2.4 Diagnostics

2.4.1 JTM Diagnostic codes

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

```
ISO 8832-JTM DEFINITIONS IMPLICIT TAGS :=
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	,
ue-no-agency-activity	[29]	NULL	,
ue-document-already-exists	[30]	NULL	,
ue-document-does-not-exist	[31]	NULL	,
ue-empty-directory	[32]	NULL	,
ue-proforma-not-present	[33]	NULL	,
ue-site-basic-class-only	[34]	NULL	,
ue-ftam-not-supported	[35]	NULL	,
ue-no-ftam-document	[36]	NULL	,
ue-ftam-disposal-error	[37]	NULL	,
w-ftam-warning	[38]	NULL	,
sf-monitor-point-storage-not-supported	[39]	NULL	,
sf-transfer-control-not-supported	[40]	NULL	,
sf-transfer-manipulation-not-supported	[41]	NULL	,
ue-commitment-and-holds-incompatible	[42]	NULL	,
ue-no-authority-to-remove-permission	[43]	NULL	,
ue-wrong-number-se-lists	[44]	NULL	,
ue-wrong-number-se-ref-lists	[45]	NULL	,
ue-wrong-number-source-lists	[46]	NULL	,
ue-wrong-number-source-ref-lists	[47]	NULL	,
ue-invalid-modification	[48]	NULL	,
ue-no-tcr-authorisation	[49]	NULL	,
ue-document-reference-not-satisfied	[50]	NULL	,
sf-implementation-failure	[51]	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”.

NOTE 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. The error code “sf-implementation-failure” is not generated by the procedures of section 3. An implementation may generate it when it is unable to follow those procedures fully.

NOTE 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 2.4.2.

2.4.2 CCR diagnostics

CCR diagnostics are carried in user data of CCR primitives (see clause 2.5), in JTM diagnostics in reports (see 2.6.7), and in embedded diagnostics in Transfer-elements (see 2.6.5).

ISO8832-JTM DEFINITIONS IMPLICIT TAGS :=

BEGIN

```

CCR-diagnostics := CHOICE
  {warning
    [0] SET OF SET
      {generator
        [0] JTM-name ,
        code
        [1] JTM-code ,
        CHOICE
          {reason
            [2] Message ,
            ftam-outcome
            [3] CHOICE
              { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
                [3] IMPLICIT ISO8571-RFCI.WriteTransferOutcome } } } ,
    - - one or more in the SET OF - - ,
  no-retry
    [1] SET OF SET
      {generator
        [0] JTM-name ,
        code
        [1] JTM-code ,

```

```

                                CHOICE
{reason                [2] Message    ,
 ftam-outcome          [3] CHOICE
  { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
    [3] IMPLICIT ISO8571-RFCI.WriteTransferOutcome } } ,
- - one or more in the SET OF - - ,
retry-later [2] SET
{retry-timer          [0] INTEGER OPTIONAL ,
 retry-reason          [1] SET OF SET
  {generator          [0] JTM-name    ,
   code               [1] JTM-code    ,
                                CHOICE
{reason                [2] Message    ,
 ftam-outcome          [3] CHOICE
  { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
    [3] IMPLICIT ISO 8571-RFCI.WriteTransferOutcome } } } ,
- - one or more in the SET OF - - } }
Basic-CCR-diagnostics := CCR-Diagnostics
( WITH COMPONENTS {warning
  ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason } ) ) }
| WITH COMPONENTS {no-retry
  ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason } ) ) }
| WITH COMPONENTS {retry-later
  ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason } ) ) }
)
END

```

NOTE The above module reference uses the definition of the syntax of FTAM control information provided in Annex B of ISO 8571-4. The datatypes 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.

2.5 User data on CCR primitives

These datatypes shall be the values used by JTM for the user data fields declared as “EXTERNAL” in ISO/IEC 9805. JTM does not use the user data fields of those CCR primitives for which no datatype is defined in this clause.

2.5.1 User data on C-BEGIN request and indication

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS :=
BEGIN
C-BEGIN-USER-DATA :=      SET
{commitment-level        [0] INTEGER (0..2)    ,
 diagnostic-code-indicator [1] SEQUENCE OF
  - - Zero, one or more in the SEQUENCE
                                Code-specifier }

Code-specifier :=
{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/IEC 8831. It shall take only the values AGENCY-ACCEPTANCE and PROVIDER-ACCEPTANCE when C-BEGIN is issued by a Basic Class implementation or on an association that is using the application context of “ISO JTM Basic Class Application-context”. Section 3 specifies the procedures related to commitment-level.

2.5.2 User data on C-READY request and indication

```
ISO8832-JTM DEFINITIONS IMPLICIT TAGS ==
BEGIN
C-READY-USER-DATA ::= SET
    { commitment-level      [0] INTEGER (0..2)
      warnings              [1] warning<CCR-diagnostics OPTIONAL ,
      accounting-info       [2] Accounting-information OPTIONAL }

Accounting-information ::= SET OF Charging-information

Charging-information ::= SEQUENCE
    { identification        [0] Identification ,
      resource-name         [1] GraphicString ,
      charging-unit         [2] GraphicString ,
      charge                [3] INTEGER      }
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. The commitment-level field can take all values, including COMPLETION.

2.5.3 User data on C-ROLLBACK request and indication

The following data type shall only be used when the C-ROLLBACK request is issued by the subordinate (the acceptor of the previous C-BEGIN indication). No datatype is defined for the user data of C-ROLLBACK request or indication when the request is issued by the superior.

```
ISO8832-JTM DEFINITIONS IMPLICIT TAGS ==
BEGIN
C-ROLLBACK-RI-USER-DATA ::= SET
    { diagnostic            [0] CHOICE
      {
          retry-later<CCR-diagnostics ,
          no-retry<CCR-diagnostics } ,
      accounting-info       [1] Accounting-information OPTIONAL }
END
```

The text of 2.5.2 shall apply to the “accounting-info” field.

2.6 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 within the “ISO JTM Basic Class Application-context”, 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) a value of the “Transfer-element” datatype; optionally followed by
- b) the presentation data values specified in a document type definition to carry the semantics of a single document.

A JTM transfer within the “ISO JTM Full Application context” consists of an indefinite number of P-DATA primitives (within a single CCR atomic action), carrying in their user data parameters the following presentation data values:

- a) a value of the “Transfer element” datatype; optionally followed by
- b) the presentation data values specified in a document type definition to carry the semantics of a single document; optionally followed by an indefinite number of
- c) an instance of the “Document-separator” datatype; followed by presentation data values as in b)

2.6.1 Top-level fields

2.6.1.1 Syntactic definitions

ISO8832-JTM DEFINITIONS IMPLICIT TAGS :=

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	,
secondary-monitoring-specification	[11] SEQUENCE OF	
	Monitoring-specification OPTIONAL	,
authorisations	[7] SET OF Authorisation-element	,
permissions	[8] SET OF Permission-element	,
accounts	[12] SET OF Account-element OPTIONAL	,
security-parameters	[13] ANY OPTIONAL	,
subjob-name-list	[9] Subjob-name-list	

(WITH COMPONENT (WITH COMPONENTS { . . . , qualifying-integer (1..MAX) }) ,
 COMPONENTS OF
 Subjob-specification
 proforma-list [10] SEQUENCE OF Proforma }

Subjob-name-list := SEQUENCE OF SEQUENCE

{proforma-name	GraphicString	,
qualifying-integer	INTEGER (0..MAX)	}

Subjob-specification := SEQUENCE

{relays	[25] SEQUENCE OF JTM-name OPTIONAL	,
target	[20] JTM-name	,
type	[21] Subjob-type	,
urgency	[22] Urgency	,
error-action	[23] Error-action	,
holds	[26] SEQUENCE OF Hold-element OPTIONAL	,
actions	[24] JTM-action-parameters	}

Subjob-type := CHOICE

{document-movement	[0] NULL	,
work-manipulation	[1] NULL	,
report-movement	[2] NULL	,
tcr-manipulation	[3] NULL	,
report-manipulation	[4] NULL	}

JTM-action-parameters := CHOICE

{document-movement	[0] Document-movement-operations	,
work-manipulation	[1] Work-manipulation-operations	,
report-movement	[2] Report-movement-operation	,
tcr-manipulation	[3] Transfer-manipulation	,
report-manipulation	[4] Report-manipulation	}

Proforma := SEQUENCE

{proforma-name	[0] GraphicString	,
spawning-control-data	[2] CHOICE	
{demand-only	[0] NULL	,
acceptance	[1] NULL	,
completion	[2] NULL	,
conditional	[3] NULL } DEFAULT completion	,
demand-spawning-handles	[3] SET OF Demand-spawning-handle OPTIONAL	,
	CHOICE	
{proforma-reference	[4] GraphicString	,
proforma-body	[1] Proforma-specification }	

```

Proforma-specification ::= SEQUENCE
{
    COMPONENTS OF Subjob-specification (WITH COMPONENTS
        { ... , actions ( WITH COMPONENTS
            { ... , report-movement ABSENT } ) } ) ,
    proforma-list [1] SEQUENCE OF Proforma
}

Basic-transfer-element ::= Transfer-element
(WITH COMPONENTS
{ ... ,
    primary-monitoring-specification (INCLUDES Basic-monitoring-specification) ,
    secondary-monitoring-specification ABSENT ,
    accounts ABSENT ,
    security-parameters ABSENT ,
    relays ABSENT ,
    type (INCLUDES Basic-subjob-type) ,
    urgency (WITH COMPONENTS {medium }) ,
    error-action (WITH COMPONENTS {terminate}) ,
    holds ABSENT ,
    actions (INCLUDES Basic-JTM-action-parameters) ,
    proforma-list ( SIZE ( 0 . 1 ) )
    ( WITH COMPONENT ( INCLUDES Basic-proforma ) ) }
)

Basic-subjob-type ::= Subjob-type
(WITH COMPONENTS { ... , ter-manipulation ABSENT ,
    report-manipulation ABSENT } )

Basic-JTM-action-parameters ::= JTM-action-parameters
(WITH COMPONENTS {
    document-movement ( INCLUDES Basic-document-movement-operations
        ( INCLUDES Basic-top-level-document-movement ) ) ,
    work-manipulation ( INCLUDES Basic-work-manipulation-operations ) ,
    report-movement ( INCLUDES Basic-report-movement-operations ) } )

Basic-proforma ::= Proforma
(WITH COMPONENTS { ... ,
    spawning-control-data ABSENT ,
    demand-spawning-handles ABSENT ,
    ( WITH COMPONENTS { proforma-body ( INCLUDES Basic-proforma-specification ) } )
} )

Basic-proforma-specification ::= Proforma-specification
(WITH COMPONENTS
{ ... ,
    relays ABSENT ,
    type ( WITH COMPONENTS { document-movement } ) ,
    urgency ( WITH COMPONENTS { medium } ) ,
    error-action ( WITH COMPONENTS { terminate } ) ,
    holds ABSENT ,
    actions ( WITH COMPONENTS
        { document-movement ( INCLUDES Basic-document-movement-operations ) }
    )
} )

Urgency ::= CHOICE
{ low [0] NULL ,
  medium [1] NULL ,
  high [2] NULL }

Error-action ::= CHOICE
{ terminate [0] NULL ,
  hold [1] INTEGER }

Demand-spawning-handle ::= GraphicString
END

```

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

Identification	2.6.4
Time-stamp	2.6.5
Audit-element	2.6.2
Monitoring-specification	2.6.3
Authorisation-element	2.6.4
Permission-element	2.6.4
Hold-element	2.6.4
Document-movement-operations	2.6.5
Work-manipulation-operations	2.6.6
Report-movement-operation	2.6.7
Report-manipulation	2.6.9
Transfer-manipulation	2.6.8

A work specification also contains

<time waiting =|>
 <estimated size =|>
 <agency activity parameters>

and, for each top-level proforma

<times spawned =|>

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 documents and document separators which are transferred as further values on P-DATA primitives. (In Basic Class, any document is an additional value on the first and only P-DATA).

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

2.6.1.2 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 2.6.2);
- b) 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;
report-manipulation	Report-manipulation;
tr-manipulation	Transfer-manipulation;
- c) 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.
- d) the elements of “authorisations”, “permissions”, “accounts” and “holds” shall all have distinct values.
- e) the “proforma-reference” in a “Proforma” must equal the “proforma-name” of either:
 - 1) the “Proforma” whose “Subjob-specification” contains the “proforma-reference”; or
 - 2) one of the “Proforma” in the same “proforma-list” as the “Proforma” in 1) above;

The following restrictions apply during the transfer of a “Transfer-element” when the association-context is “ISO JTM Basic Class Application-context”:

- a) the “Transfer-element” shall be a “Basic-transfer-element”;
- b) there shall be no “embedded” choice in any “Document-pointer” in any proforma in the “Transfer-element” (see 2.6.5)

2.6.2 Audit elements

2.6.2.1 Syntactic definitions

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

2.6.2.2 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.

2.6.3 Monitoring specifications

2.6.3.1 Syntactic definitions

```
ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=
BEGIN
Monitoring-specification ::= SEQUENCE
    {monitor-specification [0] Monitor-spec ,
     report-selector       [1] Report-selector }

Report-selector ::= BIT STRING
    {normal-termination      (0) ,
     manipulation-termination (1) ,
     abnormal-termination    (2) ,
     user-message            (3) ,
     creation                 (4) ,
     transfer                 (5) ,
     spawning                 (6) ,
     agency-acceptance        (7) ,
     modification             (8) ,
     error-diagnostic          (9) ,
     no-progress               (10) ,
     accounting-data          (11) ,
     not-supported-termination (12) ,
     violation-attempt         (13) ,
     warning-report           (14) }

Monitor-spec ::= SEQUENCE
    {monitor-relays          [20] SEQUENCE OF JTM-name OPTIONAL ,
     monitor-system-name     [0] JTM-name ,
     disposal-instructions   CHOICE
        {disposal-data       [1] SEQUENCE
            {agency           [0] jtm-se<Se-identification ,
              doc-name        [1] jtm-write-data<Document-se-reference },
              keep            [2] NULL } }

Basic-monitoring-specification ::= Monitoring-specification
    ( WITH COMPONENTS { monitor-specification ( WITH COMPONENTS
        { monitor-relays ABSENT ,
          monitor-system-name ,
          disposal-instructions ( WITH COMPONENTS
              { ... , disposal-data ( WITH COMPONENTS
                  { agency ( WITH COMPONENTS { se-name } ) ,
```

```

doc-name ( WITH COMPONENTS
            { . . . , se-access-parameter add<Se-access-parameter } )
        ) } } } } ,
report-selector } )
END

```

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

Se-identification	2.6.5
Document-se-reference	2.6.5

2.6.3.2 *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. In a “report-selector” in a “Basic-monitoring-specification”, a bit in the bit-string shall not be set to 1 unless it is one of the distinguished bits “normal-termination”, “manipulation-termination”, “abnormal-termination” or “user-message”.

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.

2.6.4 Authorisation, permission, account and hold elements

2.6.4.1 *Syntactic definitions*

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ==
BEGIN
Authorisation-element ::= SEQUENCE
    { id [0] Identification ,
      validation [1] CHOICE
        { checked-index [0] INTEGER (1..MAX) ,
          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 } }

Account-element ::= Authorisation-element

Hold-element ::= SEQUENCE
    { location [0] JTM-name ,
      CHOICE
        { reason [1] Message ,
          diagnostic-info [2] Diagnostic-information } ,
      release-permission [3] Identification OPTIONAL ,
      release-time CHOICE
        { date-time [4] GeneralizedTime ,
          time-of-day [5] NumericString (SIZE ( 4 ) ) ,
          time-interval [6] INTEGER } }

END

```

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

Diagnostic-information	2.6.7
------------------------	--------------

2.6.4.2 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”.

2.6.5 Document movement operations

2.6.5.1 Syntactic definitions

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=
BEGIN
Document-movement-operations ::= SEQUENCE OF Document-movement
Document-movement ::= SEQUENCE
    {type                [0] OBJECT IDENTIFIER
    type-parameters      [4] ANY DEFINED BY type OPTIONAL
                        - - to be filled by the ASN.1 type defined for PARAMETERS in
                        - - the document type register entry identified by “type”
    ses                  [1] SEQUENCE OF Se-identification ,
    document-block       CHOICE
        {single-form      [2] Single-form ,
        multiple-form     [3] Multiple-form }}

Se-identification ::= CHOICE
    {jtm-se              [0] SEQUENCE
    {se-name             [0] Agency-name ,
    agency-parameter     [1] Agency-parameter DEFAULT none ,
    agency-activity-label [2] GraphicString OPTIONAL ,
    se-prefix            [3] Name-list-OPTIONAL ,
    additional-authorisations [4] Additional-authorisations OPTIONAL },
    ftam-sink            [1] FTAM-name }

Agency-parameter ::= CHOICE
    {none                [0] NULL ,
    store                [1] NULL ,
    agency-format        [2] GraphicString }

Additional-authorisations ::= SEQUENCE
    {agency-password     [0] GraphicString OPTIONAL ,
    agency-account       [1] Account-element OPTIONAL ,
    agency-authorisation [2] Authorisation-element OPTIONAL }

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] Se-access-parameter }
    ftam-write-data      [1] [1] IMPLICIT ISO8571-RFCI.WriteTransferSpecification }

```

Se-access-parameter ::= CHOICE

{normal	[0] NULL	,
add	[1] NULL	,
new	[2] NULL	,
old	[3] NULL	,
append	[4] 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] Embedded-diagnostics	,
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".

Embedded-diagnostics ::= CHOICE

{embed	[0] NULL	,
error	[1] NULL	}

Source-identification ::= CHOICE

{provider	[0] NULL	,
jtm-source	[1] SEQUENCE	
{source-name	[0] Agency-name	
agency-activity-label	[20] GraphicString OPTIONAL	,
agency-parameter	[21] Agency-parameter OPTIONAL	,
additional-authorisations	[22] Additional-authorisations OPTIONAL	}
ftam-source	[21] FTAM-name	}

Document-source-reference ::= CHOICE

{jtm-read-data	[0] SEQUENCE	
{document-name	[0] Name-list	,
source-access-parameter	[1] Source-access-parameter OPTIONAL	}
ftam-read-data	[1] [0] IMPLICIT ISO8571-RFC1.ReadTransferSpecification	}

Time-stamp ::= CHOICE

{not-available	[0] NULL	,
time	[1] GeneralizedTime	}

Source-access-parameter ::= CHOICE

{move	[0] NULL	,
copy	[1] NULL	}

Multiple-form ::= SEQUENCE

{action-open-system	[0] JTM-name	,
doc-se-skeleton	[1] jtm-write-data<Document-se-reference	,
source	[2] jtm-source<Source-identification	,
doc-source-skeleton	[3] jtm-read-data<Document-source-reference	,
embed-diagnostics	[4] Embedded-diagnostics	,
document-selector	[5] CHOICE	
{empty	[0] NULL	,
local-string	[1] GraphicString	}

```

Basic-document-movement-operations ::= Document-movement-operation
    ( SIZE ( 1 ) )
    ( WITH COMPONENT ( INCLUDES Basic-document-movement ) )

```

```

Basic-document-movement ::= Document-movement
    ( WITH COMPONENTS { . . . ,
        ses ( SIZE ( 1 ) )
        ( WITH COMPONENT { jtm-se ( WITH COMPONENTS { se-name } ) } ) ,
        document-block ( INCLUDES Basic-single-form ) } )

```

```

Basic-single-form ::= [2] Single-form ( WITH COMPONENTS {
    doc-name ( WITH COMPONENTS { jtm-write-data ( WITH COMPONENTS {
        document-name ,
        se-access-parameter ( WITH COMPONENTS { normal, add } ) } ) } ) ,
    docs ( SIZE ( 1 ) ) ( WITH COMPONENT ( INCLUDES Basic-document-pointer ) ) } )

```

```

Basic-document-pointer ::= Document-pointer ( WITH COMPONENTS { . . . ,
    single-document-reference ( WITH COMPONENTS { . . . ,
        source ( WITH COMPONENTS {
            provider ,
            jtm-source ( WITH COMPONENTS { source-name } ) } ) ,
        doc-name ( WITH COMPONENTS {
            jtm-read-data ( WITH COMPONENTS { document-name } ) } ) ,
        embedded-diagnostics ( WITH COMPONENTS { embed } ) } ) } )

```

```

Basic-top-level-document-movement ::= Basic-document-movement
    ( WITH COMPONENTS { . . . ,
        document-block ( WITH COMPONENTS { . . . ,
            docs ( WITH COMPONENT ( WITH COMPONENTS
                { single-document-reference ( WITH COMPONENTS { . . . ,
                    state ( WITH COMPONENTS { failed } ) } ) } ) } )
        | WITH COMPONENTS {
            type ( { iso standard 8832 document(4) text(1) }
                | { iso standard 8832 document(4) print(2) }
                | { iso standard 8832 document(4) work-display(4) } ) ,
            ses ,
            document-block ( WITH COMPONENTS { . . . ,
                docs ( WITH COMPONENT { embedded } ) } ) } )
    )

```

END

2.6.5.2 Semantic restrictions

Within the “ISO JTM Basic Class Application-context”, the number of occurrences of the “embedded” CHOICE for “Document-pointer” shall exactly equal the number of values in the P-DATA following the Basic-transfer-element. Each value is a document. (There is at most one such occurrence in the Basic Class.)

Within the “ISO JTM Full Application-context”, the number of occurrences of the “embedded” CHOICE for “Document-pointer” shall exactly equal the number of documents in the presentation data values in the P-DATA following the “Transfer-element”.

The correspondence between the “embedded” CHOICE of “Document-pointer” and the document shall be determined by the order of the “embedded” CHOICES in the abstract syntax definition of the “Transfer-element” given in 2.6 and the order of the documents in the P-DATA. The type of each document shall agree with the “Document-type” given in the “Document-movement” containing the corresponding “embedded” CHOICE of “Document-pointer”.

The “Se-identification” in the “ses” in a “Document-movement” shall either all be the CHOICE “jtm-se” or all “ftam-sink”. If they are “ftam-sink”, the “document-block” shall be a “Single-form” in which the “doc-name” “Document-se-reference” shall be the CHOICE “ftam-write-data”. “Document-se-reference” with the CHOICE “ftam-write-data” shall only be present in a “Single-form” in a “Document-movement” which has “ftam-sink” “Se-identification”.

If a “Document-pointer” contains a “Source-identification” which is the CHOICE “ftam-source”, the “doc-name” “Document-source-reference” shall be the CHOICE “ftam-read-data”.

“Document-source-reference” with the CHOICE “ftam-read-data” shall only be present in a “Document-pointer” which has “ftam-source” “Source-identification”.

In an outer level subjob specification of a work specification whose target is this JTM ASE the “se-name” in the “ses” of a “Basic-top-level-document-movement” in the single document movement shall be an instance of a GraphicString used by the receiving open system as the name of a sink or execution agency.

A “Document-pointer” which is a “single-document-reference” and which has a “state” with the value:

not-attempted NULL

and has a “source” with the value:

provider NULL

a) shall not be in an outer level “Subjob-specification” of a work specification

b) shall only be in a “Subjob-specification” in a “Proforma” if the “action-open-system” in the “single-document-reference” is equal to the “JTM-name” of the “target” of the “Subjob-specification” in whose the “proforma-list” the “Proforma” lies.

In a “Document-pointer” which is a “single-document-reference” in which “source” has the value

provider NULL

the “document-name” in the “doc-name” shall have the value

{*s*}

where *s* is a “GraphicString”

2.6.5.2.1 In a work-manipulation transfer-element proforma

The “document-movement” in a “Proforma” which is in a “Basic-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
    {se-name      n    } } ,
document-block
  {doc-name      jtm-write-data
    {document-name  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”.

2.6.5.2.2 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.

2.6.6 Work-manipulation operations

2.6.6.1 Syntactic definitions

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::= BEGIN

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

Work-operation ::= CHOICE
 {select [0] Selector ,
 kill [1] NULL ,
 stop [2] NULL ,
 modify [4] Update ,
 display CHOICE
 {brief [3] GraphicString ,
 full [5] GraphicString } }

Selector ::= SEQUENCE
 {selector-form [0] CHOICE
 {first-header-is [0] NULL ,
 contains-header [1] NULL ,
 does-not-contain-header [2] NULL ,
 header-is [3] NULL } ,
 test [1] Header-test }

Header-test ::= CHOICE
 {
 and-clause [1] SEQUENCE
 {header-one [0] Header-test ,
 header-two [1] Header-test } ,
 not-clause [2] Header-test ,
 or-clause [3] SEQUENCE
 {header-one [0] Header-test ,
 header-two [1] Header-test } ,
 success [4] NULL ,
 fail [5] NULL }

Field-test ::= CHOICE
 {osi-job-submission-system-equals [0] JTM-name ,
 initiating-identification-equals [3] Identification-value ,
 osi-jobname-equals [1] GraphicString ,
 osi-job-local-reference-equals [4] GraphicString ,
 audit-trace-equals [5] SEQUENCE OF JTM-name-value ,
 audit-trace-contains [6] JTM-name ,
 audit-trace-first-of-list-is [7] JTM-name ,
 audit-trace-last-of-list-is [8] JTM-name ,
 primary-monitor-equals [9] Monitoring-spec-value ,
 secondary-monitors-equals [10] SET OF Monitoring-spec-value ,
 secondary-monitors-contains [11] Monitoring-spec-value ,
 authorisations-equal [12] SET OF Authorisation-value ,
 authorisations-contain [13] Authorisation-value ,
 accounts-equal [14] SET OF Account-value ,
 accounts-contain [15] Account-value ,
 permissions-equal [16] SET OF Permission-value ,
 permissions-contain [17] Permission-value ,
 security-parms-equal [18] ANY ,
 subjob-name-list-equals [19] SEQUENCE OF Subjob-name-value ,
 subjob-name-list-first-of-list [20] Subjob-name-value ,
 relays-equal [21] SEQUENCE OF JTM-name-value ,

relays-contain	[22] JTM-name	,
relays-first-of-list	[23] JTM-name	,
relays-last-of-list	[24] JTM-name	,
target-equals	[25] JTM-name	,
urgency-equals	[26] Urgency	,
holds-equals	[27] SET OF Hold-element-value	,
holds-contain	[28] Hold-element-value	,
subjob-type-equals	[2] Subjob-type	,
se-lists-equal	[29] SEQUENCE OF Se-identification-value,	
se-lists-contain	[30] Se-identification-value	,
se-ref-lists-equal	[31] SEQUENCE OF Doc-se-reference-value	,
se-ref-lists-contain	[32] Doc-se-reference-value	
source-lists-equal	[33] SEQUENCE OF Source-identification-value,	
source-lists-contain	[34] Source-identification-value	,
source-ref-lists-equal	[35] SEQUENCE OF Doc-source-reference-value	,
source-ref-lists-contain	[36] Doc-source-reference-value	,
error-action-equals	[37] Error-action	,
time-waiting-equals	[38] INTEGER	,
time-waiting-gt	[39] INTEGER	,
time-waiting-ge	[40] INTEGER	,
time-waiting-lt	[41] INTEGER	,
time-waiting-le	[42] INTEGER	,
estimated-size-equals	[43] INTEGER	,
estimated-size-gt	[44] INTEGER	,
estimated-size-ge	[45] INTEGER	,
estimated-size-lt	[46] INTEGER	,
estimated-size-le	[47] INTEGER	}

Any-item ::= [APPLICATION 2] NULL

JTM-name-value ::= CHOICE {Any-item , JTM-name }

Identification-value ::= CHOICE

```

{
    Any-item ,
    open-system    [0] JTM-name ,
    authority      [1] User-identification-authority ,
    user          [2] SEQUENCE
        {authority [0] CHOICE { Any-item , User-identification-authority } ,
         id        [1] CHOICE { Any-item , User-identification } }

```

Monitoring-spec-value ::= SEQUENCE

```

{monitor-specification CHOICE { Any-item , [0] Monitor-spec-value } ,
 report-selector       CHOICE { Any-item , [1] Report-selector } }

```

Monitor-spec-value ::= SEQUENCE

```

{monitor-system-name CHOICE { Any-item , [0] JTM-name } ,
 disposal-instructions CHOICE
    {
        Any-item ,
        disposal-data [1] SEQUENCE
            { agency CHOICE { Any-item , [0] Se-identification } ,
              doc-name CHOICE { Any-item , [1] Document-se-reference } } ,
        keep          [2] NULL } }

```

Authorisation-value ::= CHOICE

```

{ Any-item,
  SEQUENCE
    { id [0] Identification-value ,
      validation CHOICE
          {
              Any-item ,
              checked-index [0] INTEGER ,
              password      [1] CHOICE
                  {
                      Any-item ,

```

unset	[0] NULL ,
graphics	[1] GraphicString ,
binary	[2] OCTET STRING } } }

Permission-value ::= Identification-value

Account-value ::= Authorisation-value

Hold-element-value ::= CHOICE

{ Any-item ,
SEQUENCE

{ location CHOICE { Any-item , [0] JTM-name } ,
CHOICE
{
reason Any-item ,
[1] Message ,
diagnostic-info [2] Diagnostic-information } ,
release-permission [3] Identification OPTIONAL,
release-time CHOICE
{
Any-item ,
date-time [4] GeneralizedTime ,
time-of-day [5] NumericString (SIZE (4)) ,
time-interval [6] INTEGER } } }

Subjob-name-value ::= SEQUENCE

{ proforma-name CHOICE { Any-item, GraphicString } ,
qualifying-integer CHOICE { Any-item , INTEGER } }

Se-identification-value ::= CHOICE

{ Any-item
jtm-se , [0] SEQUENCE
{se-name CHOICE { Any-item , [0] Agency-name } ,
agency-parameter [1] Agency-parameter OPTIONAL ,
agency-activity-label [2] GraphicString OPTIONAL ,
se-prefix [3] Name-list OPTIONAL ,
additional-authorisations [4] Additional-authorisations-value OPTIONAL } ,
ftam-sink [1] FTAM-name }

Additional-authorisations-value ::= Additional-authorisations

Doc-se-reference-value ::= CHOICE

{
Any-item ,
jtm-write-data [0] SEQUENCE
{document-name CHOICE { Any-item , [0] Name-list } ,
se-access-parameter [1] Se access-parameter OPTIONAL } ,
ftam-write-data [1] [1] IMPLICIT
ISO8571-RFCI.WriteTransferSpecification }

Source-identification-value ::= CHOICE

{
Any-item ,
provider [0] NULL ,
jtm-source [1] SEQUENCE
{source-name CHOICE { Any-item, [0] Agency-name } ,
agency-activity-label [1] GraphicString OPTIONAL,
agency-parameter [2] Agency-parameter OPTIONAL ,
additional-authorisations [3] Additional-authorisations-value OPTIONAL } ,
ftam-source [2] FTAM-name }

```

Doc-source-reference-value ::= CHOICE
{
    Any-item ,
    jtm-read-data [0] SEQUENCE
        {document-name CHOICE { Any-item, [0] Name-list} ,
         source-access-parameter [1] Source-access-parameter OPTIONAL },
    ftam-read-data [1] [0] IMPLICIT ISO8571-RFCL.ReadTransferSpecification }

Update ::= CHOICE
{primary-monitor-set-to [0] Monitoring-specification ,
 secondary-monitors-set-to [1] SET OF Monitoring-specification ,
 secondary-monitors-add [2] Monitoring-specification ,
 secondary-monitors-remove [3] Monitoring-specification-value ,
 authorisations-add [4] Authorisation-element ( WITH COMPONENTS
    { . . . , validation ( WITH COMPONENTS { . . . , checked-index ABSENT } ) } ) ,
 permissions-add [5] Permission-element ,
 permissions-remove [6] Permission-element ,
 accounts-add [7] Account-element ( WITH COMPONENTS
    { . . . , validation ( WITH COMPONENTS { . . . , checked-index ABSENT } ) } ) ,
 relays-set-to [8] SEQUENCE OF JTM-name ,
 target-set-to [9] JTM-name ,
 urgency-set-to [10] Urgency ,
 holds-set-to [11] SET OF Hold-element ( WITH COMPONENTS
    { . . . , ( WITH COMPONENTS { . . . , diagnostic-info ABSENT } ) } ) ,
 holds-add [12] Hold-element ( WITH COMPONENTS
    { . . . , ( WITH COMPONENTS { . . . , diagnostic-info ABSENT } ) } ) ,
 holds-remove [13] Hold-element-value ,
 se-lists-set-to [14] SEQUENCE OF Se-identification ,
 se-ref-lists-set-to [15] SEQUENCE OF Document-Se-reference ,
 source-lists-set-to [16] SEQUENCE OF Source-identification ,
 source-ref-lists-set-to [17] SEQUENCE OF Document-source-reference ,
 error-action-set-to [18] Error-action }

```

```

Basic-work-manipulation-operations ::= Work-manipulation-operations
    ( SIZE ( 2 ) ) ( WITH COMPONENT ( INCLUDES Basic-work-operation ) )

```

```

Basic-work-operation ::= Work-operation
    ( WITH COMPONENTS { . . . , select ( INCLUDES Basic-selector ) ,
        modify ABSENT ,
        display ( WITH COMPONENTS { brief ( "DISPLAY" ) } ) } )

```

```

Basic-selector ::= SEQUENCE
{ selector-form [0] [0] NULL ,
  test [1] [1] SEQUENCE
    { header-one [0] [0] osi-job-submission-system-equals<Field-test ,
      header-two [1] [1] SEQUENCE
        { header-one [0] [0] osi-jobname-equals<Field-test ,
          header-two [1] [0] subjob-type-equals<Field-test } } }

```

END

NOTE The datatypes used in the "Selector" are the data-structures used in the "Transfer-element" datatype, transformed as follows:

- every non-OPTIONAL embedded datatype except a CHOICE or a datatype which is directly contained in a CHOICE, is replaced by a CHOICE of "Any-item" and the datatype;
- every embedded CHOICE has an additional component of "Any-item" added.

2.6.6.2 Semantic restrictions

A "Field-test" shall not contain structured values in which all components are the "Any-item" datatype.

In a "Work-manipulation-operations", the first "Work-operations" shall be a "Work-operations (WITH COMPONENTS {select})" and the last shall not be "Work-operations (WITH COMPONENTS {select})".

The literal ANY-ITEM, when in a <selector value> in a work specification within an open system where the corresponding datatype is OPTIONAL, shall be semantically equivalent to an omitted value. On transmission the value shall be omitted. On receipt, an omitted value shall be treated as ANY-ITEM.

2.6.7 Report-movement operation

2.6.7.1 Syntactic definitions

ISO8832-JTM DEFINITIONS IMPLICIT TAGS :=

BEGIN

Report-movement-operation := SET OF SEQUENCE

{monitor-indices [0] SET OF INTEGER DEFAULT {0} ,
report [1] Single-report }

Single-report := SEQUENCE

{name-of-reporter [0] JTM-name ,
time [1] Time-stamp ,
event-identification [9] CHOICE
 {normal-termination [0] Event-parameters
 (WITH COMPONENTS { text, spawn-count }) ,
 manipulation-termination [1] Event-parameters
 (WITH COMPONENTS { text, spawn-count }) ,
 abnormal-termination [2] Event-parameters
 (WITH COMPONENTS { text, spawn-count, errors
 (WITH COMPONENT (WITH COMPONENTS { . . . ,
 diagnostics (WITH COMPONENTS { no-retry }) })) }) ,
 user-message [3] Event-parameters
 (WITH COMPONENTS { text }) ,
 creation [4] Event-parameters
 (WITH COMPONENTS {text, target-list, hold-state}) ,
 transfer [5] Event-parameters
 (WITH COMPONENTS {text, target-list, hold-state}) ,
 spawning [6] Event-parameters
 (WITH COMPONENTS {text, target-list, hold-state}) ,
 agency-acceptance [7] Event-parameters
 (WITH COMPONENTS {text}) ,
 modification [8] Event-parameters
 (WITH COMPONENTS {text, target-list, hold-state, stop-signal}) ,
 error-diagnostic [9] Event-parameters
 (WITH COMPONENTS {text, errors
 (WITH COMPONENT (WITH COMPONENTS { . . . ,
 diagnostics (WITH COMPONENTS {no-entry}) })) }) ,
 no-progress [10] Event-parameters
 (WITH COMPONENTS {text, errors
 (WITH COMPONENT (WITH COMPONENTS { . . . ,
 diagnostics (WITH COMPONENTS {no-retry}) })) }) ,
 accounting-data [11] Event-parameters
 (WITH COMPONENTS {text, accounting-information}) ,
 not-supported-termination [12] Event-parameters
 (WITH COMPONENTS {text, spawn-count}) ,
 violation-attempt [13] Event-parameters
 (WITH COMPONENTS {text, security-data}) ,
 warning-report [14] Event-parameters
 (WITH COMPONENTS {text, errors
 (WITH COMPONENT (WITH COMPONENTS { . . . ,
 diagnostics (WITH COMPONENTS {warning}) })) }) }

Subjob-identification ::= SEQUENCE

{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-parameters ::= SEQUENCE

{text	[0] Message	,
spawn-count	[1] INTEGER (0..MAX)	OPTIONAL ,
errors	[2] SET OF Diagnostic-information	OPTIONAL ,
target-list	[3] SEQUENCE	
{relays	[0] SEQUENCE OF JTM-name	OPTIONAL ,
target	[1] JTM-name }	OPTIONAL ,
security-data	[4] SEQUENCE	
{audit-trace	[0] SEQUENCE OF Audit-element ,	
	COMPONENTS OF Subjob-identification }	OPTIONAL ,
hold-state	[5] CHOICE	
{not-held	[0] NULL ,	
held	[1] NULL }	OPTIONAL ,
stop-signal	[6] CHOICE	
{stopped	[0] NULL ,	
modified	[1] NULL }	OPTIONAL ,
accounting-info	[7] Accounting-information	OPTIONAL }

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] CCR-diagnostics }

Basic-report-movement-operations ::= Report-movement-operation

(WITH COMPONENT (WITH COMPONENTS
 { monitor-indices ({ 0 }) ,
 report (WITH COMPONENTS { . . . ,
 event-identification (WITH COMPONENTS {
 normal-termination,
 manipulation-termination,
 abnormal-termination,
 user-message }) }))))

END

2.6.8 Transfer-manipulation operations

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=

BEGIN

Transfer-manipulation ::= CHOICE

{set	[0] Tcr-set-operation ,
check	[1] Tcr-check-operation ,
display	[2] Tcr-display-operation }

```

Tcr-set-operation ::=          SEQUENCE
    {set-by             [0] JTM-name  ,
     recipient          [1] JTM-name  ,
     control-space      [2] Control-specification  }

Tcr-check-operation ::=       SEQUENCE
    {checking-by        [0] JTM-name  ,
     recipient          [1] JTM-name  ,
     control-spec       [2] Control-specification  }

Tcr-display-operation ::=     SEQUENCE
    { CHOICE
      {destination      [0] JTM-name  ,
       all              [1] NULL  }      ,
      doc-name          [2] Graphic-string  }

Control-specification ::=     CHOICE
    {uncontrolled       [0] NULL  ,
     selection          [1] SEQUENCE OF Selector  }
END

```

2.6.9 Report-manipulation-operations

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=
BEGIN
Report-manipulation ::= CHOICE
    {delete             [0] Report-selection  ,
     display            [1] SEQUENCE
      {selection         [0] Report-selection  ,
       docname           [1] Graphic String  } }

Report-selection ::= SEQUENCE
    {submission-constraint [0] JTM-name          OPTIONAL ,
     initiator-constraint  [1] Identification     OPTIONAL ,
     reference-constraint  [2] GraphicString       OPTIONAL ,
     jobname-constraint    [3] GraphicString       OPTIONAL ,
     subjob-constraint     [4] Subjob-name-list    OPTIONAL ,
     type-constraint       [5] Subjob-type         OPTIONAL ,
     event-constraint      [6] Report-selector     OPTIONAL ,
     initiating-time-constraint [7] SEQUENCE
      {starting-time      [0] Time-stamp ,
       finishing-time      [1] Time-stamp }      OPTIONAL }
END

```

2.7 Work-display, report-display and tcr-display documents

2.7.1 Syntactic definitions

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=
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 ,
        [1] Full-display  } }

```

Brief-display ::= SEQUENCE

```
{details [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 ,
   subjob-name-list [5] Subjob-name-list ,
   type [6] Subjob-type } ,
status [1] SET OF Destination-status }
```

Full-display ::= SEQUENCE

```
{header-list [0] SEQUENCE OF Header ,
status [1] SET OF Destination-status }
```

Header ::= SEQUENCE

```
{osi-job-submission-system [0] JTM-name ,
 initiating-identification [1] Identification ,
 initiating-time [2] Time-stamp ,
 osi-job-name [3] Graphic String ,
 osi-job-local-reference [4] Graphic String ,
 primary-monitoring-specifications [5] Monitoring-specification ,
 secondary-monitoring-specifications [6] SEQUENCE OF
   Monitoring-specification OPTIONAL,
 authorisations [7] SET OF Authorisation-element ,
 permissions [8] SET OF Permission-element ,
 accounts [9] SET OF Account-element ,
 security-parameters [10] ANY OPTIONAL ,
 subjob-name-list [11] subjob-name-list ,
 relays [12] SEQUENCE OF JTM-name OPTIONAL ,
 target [13] JTM-name OPTIONAL ,
 urgency [14] Urgency OPTIONAL ,
 holds [15] SEQUENCE OF Hold-element
   OPTIONAL ,
 error-action [16] Error-action OPTIONAL ,
 type [17] Subjob-type OPTIONAL ,
 se-list [18] SEQUENCE OF Se-identification
   OPTIONAL ,
 se-ref-list [19] SEQUENCE OF Document-se-reference
   OPTIONAL ,
 source-list [20] SEQUENCE OF Source-identification
   OPTIONAL ,
 source-ref-list [21] SEQUENCE OF
   Document-source-reference
   OPTIONAL ,
 time-waiting [22] INTEGER ,
 estimated-size [23] INTEGER }
```

Destination-status ::= SEQUENCE

```
{destination [0] CHOICE
  {recipient [0] JTM-name ,
   source-agency [1] CHOICE
     {agency [0] Agency-name ,
      ftam [1] FTAM-name } } ,
   se-agency [2] CHOICE
     {agency [0] Agency-name ,
      ftam [1] FTAM-name } } ,
status [1] CHOICE
  {in-progress [0] Message ,
```

```

        accepted          [1] Message ,
        waiting           [2] SEQUENCE
        {
            [0] Message
            [1] retry-later<CCR-diagnostics OPTIONAL } },
    time                  [2] Time-stamp }

```

Basic-work-display ::= [APPLICATION 4] SEQUENCE OF SEQUENCE

```

{ displaying-system      [0] JTM-name ,
  time                  [1] Time-stamp ,
  display               [2] SEQUENCE OF [0] SEQUENCE
    { details           [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 ,
        subjob-name-list               [5] Subjob-name-list ,
        type                           [6] Subjob-type } ,
      status                         [1] SET OF SEQUENCE
        { destination              [0] CHOICE
          { recipient              [0] JTM-name ,
            source-agency          [1] [0] Agency-name ,
            se-agency              [2] [0] Agency-name } ,
          status                   [1] CHOICE
            { in-progress          [0] Message ,
              accepted             [1] Message ,
              waiting              [2] SEQUENCE
                {
                  [0] Message ,
                  [1] retry-later<CCR-diagnostics OPTIONAL } },
            time                  [2] Time-stamp } } }

```

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

```

Report-display ::= SEQUENCE
{ monitor-system      [0] JTM-name ,
  time-of-display      [1] Time-stamp ,
  [2] SEQUENCE OF Single-report }

```

Tcr-display-doc ::= [APPLICATION 6] SEQUENCE OF SEQUENCE

```

{ displaying-system      [0] JTM-name ,
  time-of-display        [1] Time-stamp ,
  controls               [2] SEQUENCE OF Transfer-control-record }

```

Transfer-control-record ::= SEQUENCE

```

{ set-by                [0] JTM-name ,
  time                  [1] Time-stamp ,
  recipient              [2] JTM-name ,
  control-spec          [3] Control-specification }

```

END

2.7.2 Semantic restrictions

A “status” of “accepted” shall not occur unless the “destination” is “se-agency” and “se-agency” is “agency”.

2.8 Summary of datatypes

In this summary, datatypes are in alphabetical order and are separated by blank lines. Where conflicts arise between this clause and earlier clauses, the earlier clauses take precedence.

```

ISO8832-JTM DEFINITIONS IMPLICIT TAGS ::=
BEGIN

Account-element ::= Authorisation-element

Account-value ::= Authorisation-value

Accounting-information ::= SET OF Charging-information

Additional-authorisations ::= SEQUENCE
{agency-password      [0] GraphicString OPTIONAL ,
 agency-account       [1] Account-element OPTIONAL ,
 agency-authorisation [2] Authorisation-element OPTIONAL }

Additional-authorisations-value ::= Additional-authorisations

Agency-name ::= GraphicString

Agency-parameter ::= CHOICE
{none                [0] NULL ,
 store              [1] NULL ,
 agency-format      [2] GraphicString }

Any-item ::= [APPLICATION 2] NULL

Application-entity-title ::= ACSE-1.AE-title

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

Authorisation-element ::= SEQUENCE
{id                [0] Identification ,
 validation        [1] CHOICE
 {checked-index    [0] INTEGER (1..MAX) ,
  password         [1] CHOICE
  {unset           [0] NULL ,
   graphics        [1] GraphicString ,
   binary          [2] OCTET STRING }}}

Authorisation-value ::= CHOICE
{Any-item,
 SEQUENCE
 {id                [0] Identification-value ,
  validation        [1] CHOICE
  {
    Any-item ,
    checked-index [0] INTEGER ,
    password      [1] CHOICE
    {
      Any-item ,
      unset      [0] NULL ,
      graphics   [1] GraphicString ,
      binary     [2] OCTET STRING }}}}}

Basic-CCR-diagnostics ::= CCR-Diagnostics
( WITH COMPONENTS {warning
 ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason} ) ) }
 | WITH COMPONENTS {no-retry
 ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason} ) ) }
 | WITH COMPONENTS {retry-later
 ( WITH COMPONENT ( WITH COMPONENTS { . . . , reason} ) ) }
 )

```

```

Basic-JTM-action-parameters := JTM-action-parameters
    ( WITH COMPONENTS {
        document-movement ( INCLUDES Basic-document-movement-operations
            ( INCLUDES Basic-top-level-document-movement ) ) ,
        work-manipulation ( INCLUDES Basic-work-manipulation-operations ) ,
        report-movement ( INCLUDES Basic-report-movement-operations ) } )

Basic-document-movement := Document-movement
    ( WITH COMPONENTS { ... ,
        ses ( SIZE ( 1 ) )
        ( WITH COMPONENT { jtm-se ( WITH COMPONENTS { se-name } ) } ) ,
        document-block ( INCLUDES Basic-single-form ) } )

Basic-document-movement-operations := Document-movement-operation
    ( SIZE ( 1 ) )
    ( WITH COMPONENT ( INCLUDES Basic-document-movement ) )

Basic-document-pointer := Document-pointer ( WITH COMPONENTS { ... ,
    single-document-reference ( WITH COMPONENTS { ... ,
        source ( WITH COMPONENTS {
            provider ,
            jtm-source ( WITH COMPONENTS { source-name } ) } ) ,
        doc-name ( WITH COMPONENTS {
            jtm-read-data ( WITH COMPONENTS { document-name } ) } ) ,
        embedded-diagnostics ( WITH COMPONENTS { embed } ) } ) } )

Basic-monitoring-specification := Monitoring-specification
    ( WITH COMPONENTS { monitor-specification ( WITH COMPONENTS
        { monitor-relays ABSENT ,
            monitor-system-name ,
            disposal-instructions ( WITH COMPONENTS
                { ... , disposal-data ( WITH COMPONENTS
                    { agency ( WITH COMPONENTS { se-name } ) ,
                    doc-name ( WITH COMPONENTS
                        { ... , se-access-parameter add<Se-access-parameter } ) } ) } ) ,
            report-selector } )

Basic-proforma := Proforma
    ( WITH COMPONENTS { ... ,
        spawning-control-data ABSENT ,
        demand-spawning-handles ABSENT ,
        ( WITH COMPONENTS { proforma-body ( INCLUDES Basic-proforma-specification ) } )
    } )

Basic-proforma-specification := Proforma-specification
    ( WITH COMPONENTS
        { ... ,
            relays ABSENT ,
            type ( WITH COMPONENTS { document-movement } ) ,
            urgency ( WITH COMPONENTS { medium } ) ,
            error-action ( WITH COMPONENTS { terminate } ) ,
            holds ABSENT ,
            actions ( WITH COMPONENTS
                { document-movement ( INCLUDES Basic-document-movement-operations ) }
            } )
    )

```

```

Basic-report-movement-operations := Report-movement-operation
( WITH COMPONENT ( WITH COMPONENTS
    { monitor-indices ( { 0 } ) ,
      report ( WITH COMPONENTS { . . . ,
        event identification ( WITH COMPONENTS {
          normal-termination,
          manipulation-termination,
          abnormal-termination,
          user-message } ) } ) } ) )

Basic-selector := SEQUENCE
{ selector-form [0] [0] NULL ,
  test [1] [1] SEQUENCE
    { header-one [0] [0] osi-job-submission-system-equals<Field-test ,
      header-two [1] [1] SEQUENCE
        { header-one [0] [0] osi-jobname-equals<Field-test ,
          header-two [1] [0] subjob-type-equals<Field-test } } }

Basic-single-form := [2] Single-form ( WITH COMPONENTS {
  doc-name ( WITH COMPONENTS { jtm-write-data ( WITH COMPONENTS {
    document-name ,
    se-access-parameter ( WITH COMPONENTS { normal, add } ) } ) } ) ,
  docs ( SIZE ( 1 ) ) ( WITH COMPONENT ( INCLUDES Basic-document-pointer ) ) } )

Basic-subjob-type := Subjob-type
( WITH COMPONENTS { . . . , tcr-manipulation ABSENT ,
  report-manipulation ABSENT } )

Basic-top-level-document-movement := Basic-document-movement
( WITH COMPONENTS { . . . ,
  document-block ( WITH COMPONENTS { . . . ,
    docs ( WITH COMPONENT ( WITH COMPONENTS
      { single-document-reference ( WITH COMPONENTS { . . . ,
        state ( WITH COMPONENTS { failed } ) } ) } ) } )
  | WITH COMPONENTS {
    type ( { iso standard 8832 document(4) text(1) }
      | { iso standard 8832 document(4) print(2) }
      | { iso standard 8832 document(4) work-display(4) } ) ,
    ses ,
    document-block ( WITH COMPONENTS { . . . ,
      docs ( WITH COMPONENT { embedded } ) } ) } )

Basic-transfer-element := Transfer-element
( WITH COMPONENTS
  { . . . ,
    primary monitoring-specification (INCLUDES Basic-monitoring-specification) ,
    secondary-monitoring-specification ABSENT ,
    accounts ABSENT ,
    security-parameters ABSENT ,
    relays ABSENT ,
    type (INCLUDES Basic-subjob-type) ,
    urgency (WITH COMPONENTS {medium } ) ,
    error-action (WITH COMPONENTS {terminate} ) ,
    holds ABSENT ,
    actions (INCLUDES Basic-JTM-action-parameters) ,
    proforma-list ( SIZE ( 0 . . 1 ) )
      ( WITH COMPONENT ( INCLUDES Basic-proforma ) )
  )

```

Basic-work-display ::= [APPLICATION 4] SEQUENCE OF SEQUENCE

```
{ displaying-system  [0] JTM-name ,
  time               [1] Time-stamp ,
  display             [2] SEQUENCE OF [0] SEQUENCE
    { details        [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 ,
        subjob-name-list          [5] Subjob-name-list ,
        type                      [6] Subjob-type } ,
      status                     [1] SET OF SEQUENCE
        { destination            [0] CHOICE
          { recipient            [0] JTM-name ,
            source-agency        [1] [0] Agency-name ,
            se-agency            [2] [0] Agency-name } ,
          status                 [1] CHOICE
            { in-progress        [0] Message ,
              accepted           [1] Message ,
              waiting            [2] SEQUENCE
                {
                  [0] Message ,
                  [1] retry-later<CCR-diagnostics OPTIONAL } } ,
            time                 [2] Time-stamp } } }
```

Basic-work-manipulation-operations ::= Work-manipulation-operations
(SIZE (2)) (WITH COMPONENT (INCLUDES Basic-work-operation))

Basic-work-operation ::= Work-operation
(WITH COMPONENTS { ... , select (INCLUDES Basic-selector) ,
modify ABSENT ,
display (WITH COMPONENTS { brief ("DISPLAY") }) })

Brief-display ::= SEQUENCE
{ details [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 ,
 subjob-name-list [5] Subjob-name-list ,
 type [6] Subjob-type } ,
 status [1] SET OF Destination-status }

C-BEGIN-USER-DATA ::= SET
{ commitment-level [0] INTEGER (0 .. 2) ,
 diagnostic-code-indicator [1] SEQUENCE OF
 - - Zero, one or more in the SEQUENCE
 Code-specifier }

C-READY-USER-DATA ::= SET
{ commitment-level [0] INTEGER (0 .. 2)
 warnings [1] warning<CCR-diagnostics OPTIONAL ,
 accounting-info [2] Accounting-information OPTIONAL }

C-ROLLBACK-RI-USER-DATA ::= SET
{ diagnostic [0] CHOICE
 {
 retry-later<CCR-diagnostics ,
 no-retry<CCR-diagnostics } ,
 accounting-info [1] Accounting-information OPTIONAL }

```

CCR-diagnostics ::=      CHOICE
{warning            [0] SET OF SET
  {generator        [0] JTM-name   ,
   code             [1] JTM-code   ,
                   CHOICE
   {reason          [2] Message    ,
    ftam-outcome     [3] CHOICE
      { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
        [3] IMPLICIT ISO8571-RFCI.WriteTransferOutcome } } } ,
  - - one or more in the SET OF - - ,

no-retry            [1] SET OF SET
  {generator        [0] JTM-name   ,
   code             [1] JTM-code   ,
                   CHOICE
   {reason          [2] Message    ,
    ftam-outcome     [3] CHOICE
      { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
        [3] IMPLICIT ISO8571-RFCI.WriteTransferOutcome } } } ,
  - - one or more in the SET OF - - ,

retry-later         [2] SET
  {retry-timer      [0] INTEGER OPTIONAL ,
   retry-reason     [1] SET OF SET
     {generator      [0] JTM-name   ,
      code           [1] JTM-code   ,
                      CHOICE
      {reason        [2] Message    ,
       ftam-outcome   [3] CHOICE
         { [2] IMPLICIT ISO8571-RFCI.ReadTransferOutcome ,
           [3] IMPLICIT ISO8571-RFCI.WriteTransferOutcome } } } ,
  - - one or more in the SET OF - - } }

Charging-information ::= SEQUENCE
{identification     [0] Identification ,
 resource-name      [1] GraphicString  ,
 charging-unit      [2] GraphicString  ,
 charge             [3] INTEGER       }

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

Control-specification ::= CHOICE
{uncontrolled       [0] NULL           ,
 selection          [1] SEQUENCE OF Selector }

Demand-spawning-handle ::= GraphicString

Destination-status ::= SEQUENCE
{destination        [0] CHOICE
  {recipient         [0] JTM-name   ,
   source-agency     [1] CHOICE
     {agency         [0] Agency-name ,
      ftam [1] FTAM-name } } ,
  se-agency          [2] CHOICE
    {agency          [0] Agency-name ,
     ftam            [1] FTAM-name   } ,
 status             [1] CHOICE
  {in-progress       [0] Message    ,

```