ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-131

REFERENCED DATA TRANSFER

July 1988

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

STANDARD ECMA-131

REFERENCED DATA TRANSFER

July 1988

BRIEF HISTORY

This ECMA Standard is one of a set of standards for Distributed Office Applications.

Distributed-office-applications are used by an integrated distributed office system, consisting of user nodes and server nodes linked by a network. The user nodes access the server nodes via the network, using access protocols.

In such an environment, data processing applications that within a single host act as a single piece, have been split among the different intelligent components of the system. This splitting has led to the need for standardization of inter-relationships between the different parts of an application.

In this environment the distributed-office-applications should satisfy the following objectives:

- Make easier the implementation of application-processes developed for a distributed environment based on micro-processors and large or medium sized mainframes interconnected through local area network or wide area network means;
- Reduce the processing delay time for document related activities such as document filing and retrieval, document distribution, printing, etc., and group communication related activities such as interpersonal messaging, user directory and authentication processes, etc.;
- Allow concurrent processing of different tasks within the distributed office system;
- Reduce the overall size of an office system and facilitate its modular extension.

Within Distributed Office Applications, there will be applications that will act as a Source or Sink of objects whose values are of comparatively large quantities, for example files, documents or body parts.

The transfer of large data object values conceptually involves three parties: an Initiator which requests the transfer, a Source which produces the data object value and a Sink which consumes the data object value. To achieve economies in the use of transmission facilities and hence more efficient use of system resources, a mechanism is required whereby a Source can provide an Initiator with a reference to a data object value. This reference can then be given by the Initiator to the Sink, which can directly contact the Source to obtain the data object value. This mechanism is known as Referenced Data Transfer Facility. The characteristics of the Referenced Data Transfer Facility provide a mechanism to perform the Referenced Data Transfer between an open ended list of parties.

Adopted as an ECMA Standard by the General Assembly of 30th June 1988.

TABLE OF CONTENTS

	Page
1. SCOPE AND FIELD OF APPLICATION	1
2. REFERENCES	1
3. DEFINITIONS AND TERMINOLOGY	1
3.1 General Terminology	1
3.2 Specific Terminology	2
3.2.1 Referenced Data Transfer	2
3.2.2 Produce-operation	2
3.2.3 Consume-operation	2
3.2.4 Initiator-application-process	3
3.2.5 Source-application-process	3
3.2.6 Sink-application-process	3
3.2.7 Transfer-operation	3
3.2.8 Data object value	3
3.2.9 RDT-reference	3
3.3 Acronyms	3
3.4 Conventions	3
4. REFERENCED DATA TRANSFER FACILITY	4
4.1 Functional Model	4
4.2 Architectural Model	6
4.3 Generic Operations	6
4.4 Specific Operations	6
5. RDT REFERENCE	7
5.1 RDT-reference Structure	7
5.1.1 Logical-identifier	8
5.1.2 Locational-identifier	8
5.1.3 Local-reference	8
5.1.4 Quality-of-service	8
5.1.5 Data-object-type	9
5.1.6 Token	10
5.2 Abstract Syntax	10
6. RDT SERVICES AND PROTOCOL	12
6.1 Service Element Description	12
6.1.1 Transfer-operation	12
6.1.2 Extend-operation	12

6.1.3 Value-not-available Error 13
6.1.4 Value-altered Error 13
6.1.5 Extend-rejected Error 13
6.1.6 Unknown-reference Error 13
6.1.7 Access-denied Error 13

6.2 Abstract Syntax and Protocol 13

7. RDT CONTEXTS AND PROTOCOL 15

7.1 Overview 15
7.2 Application Contexts 16
7.3 Bind and Unbind Operations 18
7.4 Remote Operational Priorities 18

8. CONFORMANCE 19

8.1 General 19
8.2 Equipment 19
8.3 Peer Equipment 19
8.4 Conformance to the RDT Protocol 19

APPENDIX A - LIST OF OBJECT IDENTIFIER VALUES ASSIGNED IN THIS STANDARD 21

APPENDIX B - IMPACT ON PROTOCOL STANDARDS USING RDT 23

1. SCOPE AND FIELD OF APPLICATION

This ECMA Standard defines the elements used in the specification of the Referenced Data Transfer facility within Distributed Office Applications. Its content is grouped into six major groups:

- An introductory part in which references, definitions and abbreviations are collected together.
- A description of the Referenced Data Transfer facility together with a Functional model.
- Referenced Data Transfer Reference structure and Abstract Syntax.
- Referenced Data Transfer Service Element Description and Abstract Syntax.
- Referenced Data Transfer Context definition.
- Conformance requirements

An Appendix describes the impact of using RDT on the definition of other protocols.

2. REFERENCES

ECMA TR/42	Framework for Distributed Office Applications
ISO 7498	Information processing systems - Open Systems Interconnection - Basic Reference Model
ISO 8649	Information processing systems - Open Systems Interconnection - Service definition for the Association Control Service Element
ISO 8650	Information processing systems - Open Systems Interconnection - Protocol Specification for Association Control
ISO 8822	Information Processing Systems - Open Systems Interconnection - Connection Oriented Presentation Service Descriptions
ISO 8824	Specification of Abstract Syntax Notation One (ASN.1)
ISO 8825	Specification of Basic Encoding Rules for Abstract Syntax Notation one (ASN.1)
ISO 9066	Information processing systems - Text Communication - Reliable Transfer Part 1 and Part 2
ISO 9072	Information processing systems - Text Communication - Remote Operations Part 1 and Part 2

3. DEFINITIONS AND TERMINOLOGY

3.1 General Terminology

The following terms are used with the meanings defined in ISO 7498:

Application Layer
application-process

application-entity
Application-entity Title
application-service-element
Presentation Layer
presentation-connection
presentation address
protocol
service definition
transfer syntax

The following terms are used with the meanings defined in ISO 8822:

abstract syntax
abstract syntax name
transfer syntax name

The following terms are used with the meanings defined in ISO 8649:

Association Control Service Element
association-initiator
association-responder
application context

The following terms are used with the meanings defined in ISO 8824:

macro
macro notation

Coordinated Universal Time

The following term is used with the meaning defined in ISO 9066:

Reliable Transfer Service Element

The following terms are used with the meanings defined in ISO 9072:

Remote Operations
RO-notation
Remote Operation Service Element

3.2 Specific Terminology

3.2.1 Referenced Data Transfer

An information transfer from a source-application-process to a sink-application process acting cooperatively on instructions from an initiator-application-process.

3.2.2 Produce-operation

A generic class of operation from Initiator to Source.

3.2.3 Consume-operation

A generic class of operation from Initiator to Sink.

3.2.4 Initiator-application-process; Initiator

An application process which requires a Referenced Data Transfer to take place.

3.2.5 Source-application-process; Source

An application-process which has the purpose to supply a data object value which is currently available to it.

3.2.6 Sink-application-process; Sink

An application-process which makes use of a data object value which is being supplied to it.

3.2.7 Transfer-operation

An operation which results in the transfer of a data object value from the Source to the Sink.

3.2.8 Data object value

The data object value is a value derived from a data object in accordance with a set of rules, or in the absence of any such rules, the value of the entire data object.

3.2.9 RDT-reference

Information providing a globally unique reference to a data object value for a period of time.

3.3 Acronyms

ACSE	Association Control Service Element
AC	Application Context
AE	Application Entity
APDU	Application Protocol Data Unit
ASE	Application Service Element
QoS	Quality of Service
ROSE	Remote Operations Service Element
RTSE	Reliable Transfer Service Element
RDT	Referenced Data Transfer
RDTSE	Referenced Data Transfer Service Element

3.4 Conventions

This ECMA standard uses the descriptive conventions listed below.

i) ASN.1 to specify the abstract-syntax of information objects,

ii) RO-notation as defined in ISO 9072-1.

4. REFERENCED DATA TRANSFER FACILITY

4.1 Functional Model

RDT addresses an information transfer process in which there are three parties:

- An Initiator, who requires the transfer to take place
- A Source, where the information is currently held
- A Sink, to which the information is to be transferred

In direct value transfers, the Initiator of the transfer is co-located (specially and/or temporally) with either the Source or Sink. The Referenced Data Transfer functional model describes the situation which applies when Initiator, Source and Sink are separated either specially, eg. on three different end systems, or temporally, eg. the Initiator system may at some later point in time act as the Source or Sink.

The Initiator uses an application-specific protocol to select a data object value (the complete data object, or some subset or derivative of the data object) from a Source. This class of operation is termed "produce- operation".

In the case of direct value transfer, the Source returns the data object value to the Initiator, and the Initiator takes the role of the Sink.

In the RDT case, the Initiator specifies that he requires an RDT-reference to the data object value, rather than the actual data object value. The Source provides such an RDT-reference (which is globally unique) to the Initiator.

The Initiator can also use an application-specific protocol to cause a data object value to be transferred to a Sink. This class of operation is termed "consume operation".

In the case of direct value transfer, the Initiator also acts as the Source, and provides the data object value in the protocol.

In the RDT case, the Initiator provides an RDT-reference (previously obtained from a Source) in the consume operation. The Sink uses this RDT-reference to perform a transfer-operation on the Source, to obtain the referenced data object value.

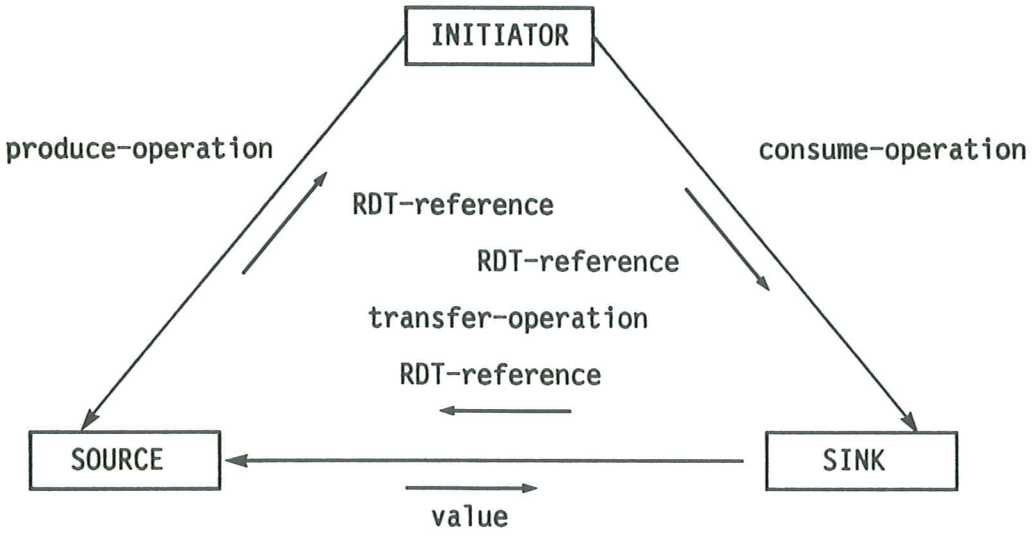

Figure 1 - Functional Model.

RDT facilities require additional functionality to be built into Source and Sink:

- the Source needs to be able to provide an RDT-reference rather than a data object value in response to a produce operation,
- the Sink needs to be able to use an RDT-reference rather than a data object value in a consume operation,
- the Sink needs to be able to invoke a transfer-operation,
- the Source needs to be able to perform a transfer-operation.

If either Source or Sink in a particular proposed transfer does not support RDT facilities, then the Initiator has no choice but to execute two consecutive direct value transfers.

In the direct value case, a produce-operation invoked by the Initiator returns a data object value from the Source to the Initiator. The Initiator transfers that data object value to the Sink in the argument of a consume-operation (note produce and consume are examples for specific operations).

In the RDT case an RDT-reference to the data object value is returned to the Initiator in the result of the produce-operation and passed to the Sink in the argument of the consume-operation. The Sink invokes a transfer-operation on the Source. The RDT-reference is contained in the argument of the transfer-operation and the RDT-referenced data object value is returned in the result of the transfer-operation.

The data object value is derived from the data object known to the Source, in accordance with application-specific rules, that are invoked by the produce-operation. An example would be where a Document Store allows access to a component of a stored document. In the absence of such selection or derivation rules, the data object value is the value of entire data object.

The RDT-reference provided by the Source may include constraints on its validity, for example it may only be valid for a limited period of time. The Sink can request an extension to the validity from the Source, but the Source is not obliged to grant such extensions.

4.2 Architectural Model

The Referenced Data Transfer facility is provided by an application-service-element, the Referenced Data Transfer Service Element RDTSE. The RDTSE is formally defined in clause 4.

The RDTSE might be included in any application context as required.

Two basic application contexts are defined in this Standard.

The default application context contains the application-service-elements ACSE, ROSE and RDTSE. The second application context contains the application-service-elements ACSE, RTSE, ROSE and RDTSE. These two application contexts are formally defined in clause 5.

4.3 Generic Operations

This standard makes use of the concept of generic operations, to describe a class of operations where specific instances are defined in protocols of other Application Standards. Two operation classes are defined:

produce-operations - A generic class of operations

Invoked by Initiator (to Source)

Invoke Selects "data object value" or "RDT-reference"

Result Returns "data object value" or "RDT-reference"

consume-operations - A generic class of operations

Invoked by Initiator (to Sink)

Invoke Supplies "data object value" or "RDT-reference"

Result has the identical semantic in both cases (In some cases, the Sink may have to wait for the result of the transfer-operation before returning the result of the consume-operation.)

Additional error information may tell the Initiator that RDT failed.

4.4 Specific Operations

Two specific operations are defined as part of the RDT facility. These are fully defined in the RDT Service Element definition.

transfer-operation

Invoked by Sink (to Source)

Invoke Specifies RDT-reference

Result Returns data object value

The transfer syntax for the data object value is determined by the Source when creating the RDT-reference. There is no possibility of the Source and Sink negotiating a transfer syntax for the data object value.

extend-operation

Invoked by Sink (to Source)

Invoke specifies RDT-reference, and requests that the validity of the RDT-reference be extended for a period of time (for example, if the Sink is unable to request transfer of the data object value at the current time).

Result returns the extension (if any) offered by the Source.

This validity of the RDT-reference is a component of the "Quality of Service" of the RDT-reference.

5. RDT REFERENCE

5.1 RDT-reference Structure

Logical-identifier	O
Locational-identifier	O
Presentation-address	M
Application-entity *	O
Application-context	D
Local-reference	M
Application	O
Specific-reference	M
Quality-of-service	O
QoS-level	O
Produce-time	M
Fidelity-time	O
Usage-of-reference	D
Data-object-type	M
Token	O

* Structure to be defined in a later version of this standard.

O optional
D *defaultable
M mandatory

Table 1 - Components of the RDT-reference

The components of the RDT-reference are described in the following clauses (see Table 1).

5.1.1 Logical-identifier

Identifies the application-entity which is part of the source-application-process. This structure can only be used if a directory server is available to allow resolution of the location of the AE, and to list the available applications contexts for the purpose of transfer. The AE is the AE containing the RDTSE.

5.1.2 Locational-identifier

Locates the application-entity which is part of the source-application-process, by means of its presentation address. The presentation-address is mandatory.

Presentation Address comprises:

Presentation selector	OPTIONAL
Session selector	OPTIONAL
Transport selector	OPTIONAL
SET OF Network-address	
Application-entity	Identifies an Application-entity.
Application-context	contains the name of the application context to which the RDTSE belongs.

5.1.3 Local-reference

This parameter is mandatory. It contains an optional application reference if more than one application is contained in an AP.

The specific-reference is the application-specific reference to the data object value.

5.1.4 Quality-of-service

If this parameter is absent, level 1 and single use of the RDT-reference is assumed.

5.1.4.1 QoS-level

If this parameter is absent, level 1 is assumed. Produce-time is mandatory for level 2 and level 3, fidelity-time is mandatory for level 3.

Produce-time is the time of execution of the produce-operation.

Fidelity-time is the time up to which the Source-application-process guarantees the RDT-reference to be valid.

In the RDT situation, three events occur in time sequence:

- execution of produce-operation (produce-time)
- execution of consume-operation (consume-time)
- execution of transfer-operation (transfer-time)

The referenced data object value as existing at produce-time may be the same at transfer-time, might be changed in between, or may not be available at all at transfer-time.

This situation can be modeled by three levels of QoS:

Level 1 — the referenced data object value might be changed or might not be available at transfer-time

Level 2 — the referenced data object value might be changed or might not be available at transfer-time. However a change which has occurred since the produce-time is indicated to the Sink and it is a decision of the Sink to process the changed data object value

Level 3 — as level 2 but now the fidelity of the RDT-reference is guaranteed up to a certain point in time (fidelity-time) (i.e. the referenced data object value is identical at produce-time and transfer-time)

The Initiator may request a certain level of QoS in the produce-operation, but the Source responds with level of QoS it can offer.

Note that in the extend-operation, the Sink may request a change of the QoS of the RDT-reference from the Source and the Source responds with a new RDT-reference containing a QoS he has determined. If the level 1 QoS is raised, the produce-time shall signify the time of the extend-operation.

5.1.4.2 Usage-of-reference

The existence of an RDT-reference requires resources in the Source. RDT provides two levels of use for an RDT-reference:

- single use
- multiple use

In the case of single use the resources required for an RDT-reference are freed after the execution of the transfer-operation. No more transfer-operations using that RDT-reference are possible.

In the case of multiple use of an RDT-reference, the resources required for an RDT-reference are kept after the transfer-operation to allow multiple transfer-operations. The lifetime of the RDT-reference is determined by the Source. However in the case of QoS level 3 the lifetime is greater or equal to the fidelity-time.

The user may request multiple use of RDT-reference in the produce-operation and the Source responds with a use-of-reference he determines.

5.1.5 Data-object-type

The type of the referenced data object is identical in produce-operations and consume- operations. This type shall be self-identifying type using the "direct reference" mechanism of ASN.1 EXTERNAL data type.

The Object Identifier within the EXTERNAL type is the value of the Data-object-type parameter in the RDT-reference. This Object Identifier identifies the abstract syntax of the data object, in combination with a specific transfer syntax.

5.1.6 Token

The token is used as an authorization mechanism whereby the Source can validate that the Sink is authorized to request the transfer of the referenced data object value.

The mechanism by which the Initiator authenticates itself to the Source is outside the scope of this Standard. Assuming that this authentication has taken place, and that the Source is satisfied that this Initiator has the right to access the specified data object value, the Source includes in the RDT-reference a token, which must be used by a Sink when requesting a transfer.

The structure of the simple-token can be as simple or complex as required by the security policy applicable to the Source and is conveyed transparently by the RDT-reference. Alternatively an externally defined token may be taken from a future Security Standard.

5.2 Abstract Syntax

The following is the formal definition of the RDT-reference, using the Abstract Syntax Notation identified in ISO 8824.

```
RDT-reference-definition        {iso identified-organization ecma rdt (# #)
                                  reference-definition (O)}

DEFINITIONS::=BEGIN

EXPORTS                         RDT-reference,
                                rdt-reference-syntax-asn1,
                                rdt-reference-abstract-syntax,
                                rdt-application-context-1, rdtx;

IMPORTS                         DistinguishedName
                                FROM InformationFramework
                                    {joint-iso-ccitt ds(5) modules(1)
                                    informationFramework(1)}

                                PresentationAddress
                                FROM SelectedAttributeTypes
                                    {joint-iso-ccitt ds(5) modules(1)
                                    selectedAttributeTypes (5)};

-- Defined Object Identifiers

 rdtx OBJECT IDENTIFIER::={iso identified-organization ecma rdt (# #)}

-- Object identifier for abstract syntax of RDT-reference in EXTERNAL

        rdt-reference-abstract-syntax OBJECT IDENTIFIER::=
                {rdtx reference-abstract-syntax (1)}
```

```
-- Object identifier for abstract syntax of RDT-reference with basic ASN.1
-- encodings in EXTERNAL

        rdt-reference-syntax-asn1 OBJECT IDENTIFIER ::=
                {rdtx reference-syntax(8) asn1(0)}

-- Object identifier for default RDT application context

        rdt-application-context-1 OBJECT IDENTIFIER::=
                {rdtx application-context-1 (2)}


-- Definition of RDT-reference type

RDT-reference::= SEQUENCE {
    ae-identifier               AE-Identifier OPTIONAL,
                                    -- AE-Identifier is mandatory except
                                    -- in some operations of RDTSE
    local-reference             Local-reference,
    data-object-type            OBJECT IDENTIFIER, -- identifying
                                    -- the abstract syntax and
                                    -- the transfer syntax of
                                    -- the referenced data value
    quality-of-service          Quality-of-Service DEFAULT{},
    token                       Token OPTIONAL}

AE-identifier::= CHOICE {
      locational-identifier [0] IMPLICIT Locational-identifier,
            logical identifier [1] IMPLICIT DistinguishedName}

Locational-identifier::= SEQUENCE {
    presentation-address            Presentation-Address,
    application-entity              [0] ANY  -- to be replaced in a
                                    -- future version of this Standard
                                    OPTIONAL,
    application-context             [1] IMPLICIT
                                    OBJECT IDENTIFIER DEFAULT
                                    {rdt-application-context-1}}

Local-reference::=SEQUENCE{
    application [0] IMPLICIT OCTET STRING OPTIONAL,
    specific-reference [1] IMPLICIT OCTET STRING}

Quality-of-Service::=SEQUENCE{
    qoS-level                   QoS-level DEFAULT level-1 NULL,
    single-use-of-reference     BOOLEAN DEFAULT TRUE}
```

```
QoS-level::=CHOICE{
                    level-1 [1] IMPLICIT NULL,
                    level-2 [2] IMPLICIT UTCTime
                        -- specifying the produce time --,
                    level-3 [3] IMPLICIT SEQUENCE{
                        produce-time        UTCTime,
                        fidelity-time       UTCTime}}

Token::=CHOICE{
            simpletoken OCTET STRING,     -- used to validate a transfer
                                          -- request which uses this
                                          -- RDT-reference
            externaltoken   EXTERNAL      -- for future proxy mechanism
            }

END
```

6. RDT SERVICES AND PROTOCOL

6.1 Service Element Description

The RDT Service Element (RDTSE) is an ASE which provides the following services:

Transfer
Extend

These are confirmed services modelled as Remote Operations between RDTSE users.

The RDTSE includes the transfer-operation and the extend-operation, and the related errors. The RDTSE is symmetric, i.e. both RDTSE users may invoke the transfer-operation and extend-operation.

6.1.1 Transfer-operation

The transfer-operation enables a Sink RDTSE user to request the Source RDTSE user to send back as a result the data object value associated with the RDT-reference supplied in the invocation.

The altered-value indication is included in the result of a transfer-operation where the returned data object value has been changed since the RDT-reference was provided by the source RDTSE user. This may occur only if the QoS of the RDT-reference is level 2, or level 3 after expiry of fidelity-time.

6.1.2 Extend-operation

The extend-operation enables a sink RDTSE user to request a source RDTSE user to change the QoS associated to the RDT-reference supplied in the invocation. This operation allows to extend either the fidelity-time or the usage-of-reference or both. The source RDTSE user responds with a new RDT-

reference which includes the requested QoS. This operation applies whatever the level of QoS associated with the RDT-reference.

6.1.3 Value-not-available Error

The value-not-available error indicates that the data object value associated with the RDT-reference is not available. This may occur only if the RDT-reference has a quality of service of level 1 or 2, and if the transfer-operation is performed after the fidelity-time is obsolete.

6.1.4 Value-altered Error

The value-altered error indicates that the data object value has been modified since the RDT-reference was issued. This may only occur if the QoS of the RDT-reference is level 2, or level 3 after expiry of the fidelity-time, and the altered-value-accepted field of the transfer-operation argument is false.

6.1.5 Extend-rejected Error

The extend-rejected error indicates that the extend-operation cannot be performed because the source RDTSE user does not offer such an operation or because the Source is not able to guarantee the requested QoS.

6.1.6 Unknown-reference Error

The unknown-reference error indicates the RDT-reference does not exist any more.

6.1.7 Access-denied Error

The access-denied error indicates the operation cannot be performed because of security reasons.

6.2 Abstract Syntax and Protocol

This is the formal definition of the RDTSE. This description is done according to ISO 8824 and ISO 9072-1.

```
RDT-service-element-definition
    {iso identified-organization ecma rdt(# #) ase-definition (3)}

DEFINITIONS::=BEGIN

EXPORTS   rDTSE, Requested-QoS-level, Altered-value, rdt-as0;


IMPORTS   RDT-reference, rdtx
          FROM RDT-reference-definition {iso identified-organization
               ecma rdt(# #) reference-definition (0)}


          OPERATION, ERROR
          FROM Remote-Operations-Notation {joint-iso-ccitt
                    remote-operations (4) notation (0)}
```

```
        APPLICATION-SERVICE-ELEMENT
        FROM Remote-Operations-Notation-extension
                {joint-iso-ccitt remote-operations (4)
                notation-extension (2)};

rDTSE APPLICATION-SERVICE-ELEMENT
        OPERATIONS {transfer, extend}
        ::={rdtx ase (7)}

rdt-as0 OBJECT IDENTIFIER::=
    {rdtx abstractSyntax (5) apdus0 (0) version1 (1)}
    -- this abstract syntax includes the apdus
    -- defined in module Remote-Operation-APDUs
    -- {joint-iso-ccitt remote-operations (4) apdus (1)}
    -- in ISO 9072-2 and the apdus defined in this module.
    -- this abstract syntax shall be used, if the RDTSE is
    -- included in application contexts not defined in this
    -- Standard.

transfer        OPERATION       -- invoked by Sink AP
                    ARGUMENT        Transfer-argument
                    RESULT      Transfer-result
                    ERRORS  {value-not-available, value-altered,
                                access-denied, unknown-reference}
                ::=1

Transfer-argument           ::=SEQUENCE {
        altered-value-accepted BOOLEAN DEFAULT TRUE,
                        -- shall be absent if QoS-level is level-1
        RDT-reference               -- ae-identifier shall be absent
        }

Transfer-result             ::=SEQUENCE{
        altered-value               Altered-value DEFAULT undefined,
        data-value                  EXTERNAL}

Altered-value               ::=ENUMERATED{
                                value-not-altered (1),
                                value-altered (2),
                                undefined (3) }
```

```
extend          OPERATION                          -- invoked by Sink AP
                    ARGUMENT Extend-argument
                    RESULT new-reference RDT-reference -- ae-identifier
                                                       -- shall be present
                    ERRORS {unknown-reference, extend-rejected}
                ::=2

Extend-argument         ::=SEQUENCE{
                        requested-QoS       Requested-QoS,
                                            RDT-reference -- ae-identifier
                                                  -- shall be absent
                        }

Requested-QoS           ::=SEQUENCE{
                    qoS-level Requested-QoS-level DEFAULT level-1 NULL,
  single-use-of-reference BOOLEAN DEFAULT TRUE}

Requested-QoS-level::=CHOICE{
            level-1 [1] IMPLICIT NULL,
            level-2 [2] IMPLICIT NULL,
            level-3 [3] IMPLICIT UTCTime
                    -- specifying the requested fidelity-
                    -- time
                    }

value-not-available     ERROR ::= 1
value-altered           ERROR ::= 2
access-denied           ERROR ::= 3
unknown-reference       ERROR ::= 4
extend-rejected         ERROR ::= 5

END
```

7. RDT CONTEXTS AND PROTOCOL

7.1 Overview

This section specifies the application contexts (ACs) that shall be employed in the construction of Referenced Data Transfer facilities. It uses the notation specified in ISO 9072-1.

This section covers the following topics:

a) Application contexts (7.2)

b) Bind and Unbind operations (7.3)

c) Remote Operation priorities (7.4)

Two Referenced Data Transfer ACs are defined in clause 7.2.

Note 1:

A RDTSE may also be included into an AC of a specific application.

7.2 Application Contexts

RDT-application-context-definition
{iso identified-organization ecma rdt (# #) ac-definition (4)}

DEFINITIONS::=BEGIN

IMPORTS BIND, UNBIND
FROM Remote-Operations-Notation
{join-iso-ccitt remote-operations (4) notation (0)}

APPLICATION-CONTEXT, aCSE
FROM Remote-Operations-Notation-extension
{joint-iso-ccitt remote-operations (4)
notation-extension (2)}

rdtx, rdt-application-context-1
FROM RDT-reference-definition
{iso identified-organization ecma rdt (# #)
reference-definition (0)}

rDTSE,rdt-as0
FROM RDT-service-element-definition
{iso identified-organization ecma rdt (# #)
ase-definition (3)}

rOSE
FROM Remote-Operations-APDUs {joint-iso-ccitt
remote-operations (4) apdus (1)}

rTSE
FROM Reliable-Transfer-APDUs {joint-iso-ccitt
reliable-transfer (3) apdus (1)}

Name
FROM InformationFramework {joint-iso-ccitt ds (5)
modules (1) informationFramework (1)};

```
rdt-ac-1        APPLICATION-CONTEXT
                APPLICATION SERVICE ELEMENTS {aCSE)
                BIND RDTBind
                UNBIND RDTUnbind
                REMOTE OPERATIONS {rOSE}
                OPERATIONS OF {rDTSE}
                ABSTRACT SYNTAXES {aCSE-as, rdt-as0, rdt-as1}
                ::=rdt-application-context-1

aCSE-as         OBJECT IDENTIFIER:={joint-iso-ccitt association-control
                (2) abstractSyntax (1) apdus (0) version1 (1)}
                        -- as defined in ISO 8650}

rdt-as1 OBJECT IDENTIFIER::={rdtx abstractSyntax (5) apdus1 (1)
                                        version1 (1)}
        -- this abstractSyntax includes the APDUs
        -- defined in this module (Bind and Unbind)

rdt-ac-2        APPLICATION-CONTEXT
                APPLICATION SERVICE ELEMENTS {aCSE, rTSE}
                BIND RDTBind
                UNBIND RDTUnbind
                REMOTE OPERATIONS {rOSE}
                OPERATIONS OF {rDTSE}
                ABSTRACT SYNTAXES {aCSE-as, rdt-as0, rdt-as2}
                ::={rdtx application-context-2 (6)}

rdt-as2         OBJECT IDENTIFIER::=
                {rdtx abstractSyntax (5) apdus2 (2) version1 (1)}
                -- this abstract Syntax includes the APDUs defined
                -- for rdt-as1 and the APDUs defined in module
                -- Reliable-Transfer-APDUs {joint-iso-ccitt
                -- reliable-transfer (3) apdus (0)}

RDTBind             ::=BIND
                ARGUMENT        RDTBindArgument
                RESULT          RDTBindResult
                BIND-ERROR      RDTBindError

RDTUnbind           ::=UNBIND
```

```
RDTBindArgument::=SET{
        credentials [0] CHOICE{
    unprotected-Credentials [0] IMPLICIT Simple-Credentials,
        external-Credentials [1] EXTERNAL} OPTIONAL}

Simple-Credentials::=SEQUENCE{
                user-Name       Name,
                user-Password   OCTET STRING}

RDTBindResult   ::=RDTBindArgument

RDTBindError    ::=SET{
            refuseReason [0] IMPLICIT INTEGER
                {busy (0), security-failure (1)}

-- Note: The transfer syntax name
-- {joint-iso-ccitt asn1 (1) basic-encoding (1)}
-- can be used for all abstract syntaxes listed in this module

END
```

7.3 Bind and Unbind Operations

The RDTBind and RDTUnbind operations are used by the association-initiator and association-responder at the beginning and end of a particular period of association. The binding process may include the exchange of simple authentication information whereby the association-initiator and association-responder can authenticate each other (see 7.2). There is also the possibility for use of other authentication mechanisms when these are standardized.

7.4 Remote Operational Priorities

The ROSE provides for assignment of relative priorities to the APDUs that may be exchanged over an association, and to the association's release.

The first column of Table 2 lists the priorities that shall be used for Referenced Data Transfer. (The lower the value the higher the priority).

The second column gives the particular services that shall be assigned those priorities in the context of the RDTSE.

Priority	Service
0	Association release
1	RO-REJECT-U RO-ERROR
2	RO-RESULT
3	RO-INVOKE

Table 2 - Remote Operational Priorities

8. CONFORMANCE

8.1 General

This clause defines the conformance requirements for the RDT Protocol specified in clause 6 and 7 of this Standard. Only the externally visible and externally testable criteria are defined.

8.2 Equipment

The conformance requirement is for equipment which consists of hardware and/or software and has the purpose of conforming with this Standard. The equipment may also have other purposes.

8.3 Peer Equipment

Any execution of the protocol necessarily involves a peer equipment with which the subject equipment communicates. For purposes of verifying conformance, it is assumed that this other peer equipment

- is operating in conformance with this Standard;
- may be capable of controlled deviation, in that it may be the source of deliberate protocol errors for the purpose of testing.

8.4 Conformance to the RDT Protocol

A system providing services and protocols as described in the Standard shall perform one or more of the following roles:

- Source - Sink

A system claiming conformance to this Standard in the role of a Source is required:

- to be able to accept an association established by a peer equipment.
- to be able to accept correctly formatted Transfer and Extend invocations and generate correctly formatted Results or Errors in accordance with clause 6 of this Standard.

- to be able to support at least the following features:

    locational identification
    Quality of Service level
    single use of reference

A claim of conformance to this Standard in the role of a Source is required to state which of the following features are supported by the equipment to which the claim applies:

    logical identification
    Quality of Service levels 2 and 3
    Multiple use of reference
    authorization by simple token in the RDT-reference
    authentication by simple credentials

A system claiming conformance to this Standard in the role of a Sink is required:

- to be able to identify the Source of the data object value from the locational identifier in the RDT-reference.
- to be able to establish an association to a peer equipment in accordance with clause 7 of this Standard.
- to be able to generate correctly formatted Transfer and Extend invocations and accept correctly formatted Results and Errors in accordance with clause 6 of this standard.
- to be able to support the following features:

    locational identification

A claim of conformance to this Standard in the role of a Sink is required to state which of the following features are supported by the equipment to which the claim relates.

    logical identification
    authentication by simple credentials

The support of RDT mechanisms as a Source and/or Sink shall not preclude the use of direct value transfer-operations.

Any claim of conformance to this standard must state which application context(s) are supported by the equipment to which it relates.

APPENDIX A

LIST OF OBJECT IDENTIFIER VALUES ASSIGNED IN THIS STANDARD

This Appendix is not part of the Standard.

```
{rdtx reference-definition (0)}                      -- ASN.1 module
{rdtx reference-abstract-syntax (1)}                 -- abstract syntax
{rdtx application-context-1 (2)}                     -- AC
{rdtx ase-definition (3)}                            -- ASN.1 module
{rdtx ac-definition (4)}                             -- ASN.1 module
{rdtx abstractSyntax (5) apdus0 (0) version1 (1)}    -- abstract syntax
{rdtx abstractSyntax (5) apdus1 (1) version1 (1)}    -- abstract syntax
{rdtx abstractSyntax (5) apdus2 (2) version1 (1)}    -- abstract syntax
{rdtx application-context-2 (6)}                     -- AC
{rdtx ase (7)}                                       -- ASE
{rdtx reference-syntax (8) asn1 (0)}                 -- abstract syntax
                                                     -- plus encoding

rdtx OBJECT IDENTIFIER ::=
        {iso identified-organization ecma rdt (# #)}
```

APPENDIX B

IMPACT ON PROTOCOL STANDARDS USING RDT

This Appendix is not part of the Standard.

The purpose of this Appendix is to provide guidelines for the description of operations defined in protocols of other application Standards and which may involve the use of the Referenced Data Transfer facility.

Impacts of RDT on the using protocols concerns:

- negotiation of RDT usage
- produce-operations
- consume-operations
- conformance clause for application protocol Standards

B.1 Negotiation of RDT usage

The application protocol must specify how usage of RDT by the parties is determined. Three possibilities are described below:

Mandatory support. The application Standard mandates support of RDT.

Negotiation at association establishment. Mechanisms are included in the association establishment between Initiator and Source to negotiate whether RDT is to be used.

Negotiation at produce-time. Mechanisms are included in the produce-operation to negotiate whether RDT is to be used.

B.2 Produce-operations

In a produce-operation, the Initiator needs to specify whether a data object value or an RDT-reference is required from the Source. The argument of the produce-operation must at minimum be extended with an indicator, to select RDT-reference or data object value (the default being data object value).

It is also possible to include the Requested-QoS-level definition, imported from the RDT-application-service-element-definition, to request a particular QoS. The result of the produce-operation shall include an EXTERNAL type which can contain either the data object value or the RDT-reference. There are no additional errors.

B.3 Consume-operation

In a consume-operation, the argument of the invoke shall include an EXTERNAL type to contain either the data object value or the RDT-reference. It is also possible to qualify each EXTERNAL by an "Altered-Value" element definition, imported from RDT-service-element-definition. This specifies whether the Initiator prohibits the transfer of a data object value which has changed since the produce-time. There are no changes to the result, but it is possible to include new error types from RDT-service-element-definition. However, it is not mandatory for an access protocol to include these errors if the protocol designers feel that it is sufficient just to report failure of the consume-operation without additional parameters.

B.4 Impact on Conformance Statement for Application Protocols Between Initiator and Source or Sink

A system claiming conformance to the application protocol Standard in the role of a Source is required to state for which produce-operations it supports the RDT mechanisms described in clauses 3 and 4 of this Standard.

A system claiming conformance to the application protocol Standard in the role of a Sink is required to state for which consume=operations it supports the RDT mechanisms described in clauses 3 and 4 of this Standard.

The inclusion of RDT capability into an Initiator equipment impacts the conformance requirement to the application protocol. The following points need consideration when drafting a conformance statement for an Application Protocol Standard including RDT.

- Negotiation of the usage of RDT.
- Extensions to the produce-operation: in particular, how to request RDT mechanism and convey RDT-reference.
- Extension to the consume-operation: in particular how to convey the RDT-reference.