

Information technology — Data Management —

Part 3: IRDS export/import facility

ICS 35.100.70

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW



Document Sharing Hub

National foreword

This British Standard reproduces verbatim ISO/IEC 13238-3:1998 and implements it as the UK national standard.

The UK participation in its preparation was entrusted to Technical Committee IST/40, Data management and interchange, which has the responsibility to:

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

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Standards Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Find” facility of the BSI Standards Electronic Catalogue.

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.

This British Standard, having been prepared under the direction of the DISC Board, was published under the authority of the Standards Committee and comes into effect on 15 June 1999

© BSI 05-2000

Amendments issued since publication

Amd. No.	Date	Comments

ISBN 0 580 32649 7

Contents

	Page
National foreword	Inside front cover
Foreword	v
Text of ISO/IEC 13238-3	1

INTERNATIONAL STANDARD

ISO/IEC 13238-3

First edition
1998-12-15

Information technology — Data Management —

Part 3: IRDS export/import facility

*Technologies de l'information — Gestion de données —
Partie 3: Aide export/import IRDS*



Contents

	Page
Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Definitions	2
3.1 Terms from ISO/IEC 10027 IRDS Framework	2
3.2 Terms from ISO/IEC 10646-1 Universal Multiple-Octet Coded Character Set (UCS)	2
3.3 Terms from ISO/IEC 10728 IRDS Services Interface	2
3.4 Additional definitions	2
4 Abbreviations	3
5 Conventions	4
5.1 Definition of Data in the Transfer File	4
5.2 Definition of Services	4
6 Concepts and facilities	4
6.1 Overview	4
6.2 Transfer File	4
6.3 Export/import services	6
6.4 Working Sets	6
6.5 Import Control Tables	7
6.6 Content Modules	7

		Page
7	Specification of the Transfer File	7
7.1	Character Set	7
7.2	Character Coding	7
7.3	Tokens	7
7.4	White Space	9
7.5	IRDS Transfer File Structure	10
7.6	Transfer File Header	11
7.7	Transfer File IRD definition	13
7.8	Transfer File IRD content	16
7.9	Transfer File Table Transfer Syntax	17
8	Specification of the Import Control Tables	20
8.1	Table IRDS Dictionary Import Control	20
8.2	Table IRD Object Import Control	20
8.3	Table IRD Working Set Import Control	21
9	Specification of the export/import services	23
9.1	Relationship to the IRDS Services Interface	23
9.2	Services provided	23
9.3	Selection of data to be exported	23
9.4	Service Data Types and Data Structures	24
9.5	Open IRDS Export File	25
9.6	Select For Export Service	27
9.7	Clear Export Selection Service	29
9.8	Export Working Set	30
9.9	Close IRDS Export File	33
9.10	Open IRDS Import File	34
9.11	Retrieve Import File Working Set Details	36
9.12	Import Working Set	37
9.13	Import Complete File	41

	Page
9.14 Close IRDS Import File	43
10 Conformance	44
10.1 File Format Conformance	44
10.2 File Format and Services Conformance	44
Annex A (normative) State Classes and Subclasses	45
Annex B (informative) Business Requirement	50
Annex C (informative) Example Use Of the BNF syntax for transferring columns	51
Annex D (informative) Sample Transfer File Format	53
Annex E (informative) Informative illustrations of the file structure	143
Index	147
Figure 1 — File Syntax	10
Figure 2 — Header Syntax	11
Figure 3 — IRD Definition Export Syntax	14
Figure 4 — IRD Tables	15
Figure 5 — IRD Content Syntax	16
Figure 6 — Table Transfer Syntax	18
Figure 7 — Business Data Model	54
Figure 8 — Implementation Data Model	56
Figure 9 — An example screen layout to be represented in the IRD	56
Figure 10 — File Structure as an Indented List	143
Figure 11 — Overall File Structure Diagram	145
Figure 12 — Header File Structure Diagram	145
Figure 13 — Definition and Content File Structure Diagram	146

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 13238-3 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 32, *Data management services*.

ISO/IEC 13238 consists of the following parts, under the general title *Information technology — Data Management*:

- *Part 1: Export/import framework*;
- *Part 2: SQL import/export*;
- *Part 3: IRDS export/import facility*.

Annex A forms an integral part of this part of ISO/IEC 13238. Annex B to Annex E are for information only.

Introduction

ISO/IEC 13238 is a three part standard. Part 1 defines an overall framework for Export/Import facilities.

Part 2 defines facilities for export from and import to SQL databases that conform to ISO/IEC 9075.

This part, Part 3, defines facilities for export from and import to an IRD and or an IRD Definition that conform to ISO/IEC 10728.

1 Scope

This part of ISO/IEC 13238 enables the bulk transfer of all or part of the data contained in an Information Resource Dictionary (IRD) or in an Information Resource Dictionary Definition conforming to ISO/IEC 10728:1993 IRDS Services Interface.

This part of ISO/IEC 13238 defines a format for such a transfer and also services to generate the export file and services to import the file. These services are additional to the existing services defined in the IRDS Services Interface.

This version of this part of ISO/IEC 13238 defines a limited set of Export and Import services. It is expected that implementations will also provide more sophisticated services based on the use of the Transfer File structure defined in this part of ISO/IEC 13238.

The physical way in which the transfer takes place is outside the scope of this part of ISO/IEC 13238. Each transfer may be effected in one of several ways including the physical transfer of the data using a transportable storage device.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 13238. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 13238 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 6429:1992, *Information technology — Control functions for coded character sets*.

ISO/IEC 9075:1992, *Information technology — Database languages — SQL*.

ISO/IEC 10027:1990, *Information technology — Information Resource Dictionary System (IRDS) framework*.

ISO/IEC 10646-1:1993, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*.

ISO/IEC 10728:1993, *Information technology — Information Resource Dictionary System (IRDS) Services Interface*.

ISO/IEC 13238-1: —¹⁾, *Information technology — Data Management — Part 1: Export/import framework*.

¹⁾ To be published.

3 Definitions

For the purposes of this part of ISO/IEC 13238, the following definitions apply.

3.1 Terms from ISO/IEC 10027 IRDS Framework

3.1.1 Application level

3.1.2 IRD

3.1.3 IRD definition level

3.1.4 IRD definition schema level

3.1.5 IRD level

3.1.6 IRD schema

3.1.7 IRDS

3.2 Terms from ISO/IEC 10646-1 Universal Multiple-Octet Coded Character Set (UCS)

3.2.1 Basic Multilingual Plane (BMP)

3.2.2 Character

3.3 Terms from ISO/IEC 10728 IRDS Services Interface

3.3.1 IRD Object

3.3.2 Name

3.3.3 Object

3.3.4 Object Version

3.3.5 Variation Name

3.4 Additional definitions

3.4.1

clear text file encoding

a class of techniques for representing data based on first defining a human readable representation using some specific character repertoire and then defining an encoding for that repertoire

3.4.2

export IRD

the export of data from an origin or source IRDS into a Transfer File

3.4.3

export process

the process of generating a Transfer File from a source environment

3.4.4**exporter**

the agent of the export process

3.4.5**import IRD**

the import of data to a destination or target IRDS from an Transfer File

3.4.6**import process**

the process of incorporating the content of a Transfer File into a target environment

3.4.7**importer**

the agent of the import process

3.4.8**IRDS-data**

a collection of data comprising the content of all or part of an IRD Definition and/or an IRD. This is a generic term used for convenience to describe the content of a Transfer File

3.4.9**IRDS transfer file**

a Transfer File containing data which defines and describes the content of an IRD or a subset of an IRD. It is made up of three components: Transfer File Header component, Transfer File IRD Definition component, and Transfer File IRD Content component

3.4.10**transfer file**

a file containing data to be interchanged. It is made up of a header, and a number of components. Components contain either data, or data definition data

3.4.11**transfer file header**

the first component of a Transfer File. This component contains data that uniquely defines the Transfer File. It contains details of the export IRD identification and time stamp

3.4.12**transfer file IRD definition**

the second (and optional) component of a Transfer File. This component contains the IRD definition that defines the following IRD content

3.4.13**transfer file IRD content**

the third (and optional) component of a Transfer File. This component contains the IRD content to be interchanged

4 Abbreviations

Abbreviations used in this part of ISO/IEC 13238 are:

- | | | |
|----|------|---|
| a) | BMP | Basic Multilingual Plane; |
| b) | BNF | Bachus Naur Form; |
| c) | IRDS | Information Resource Dictionary System; |
| d) | IRD | Information Resource Dictionary; |
| e) | TF | Transfer File; |
| f) | UTC | Co-ordinated Universal Time. |

5 Conventions

5.1 Definition of Data in the Transfer File

An extended Bachus Naur Form (BNF) is used to describe the sequence of data in the Transfer File. The form of BNF used is that defined in ISO/IEC 9075, **3.2**.

The names of the syntax units start with a lower case alphabetic character and intermediate characters are capitalised when they start a word.

5.2 Definition of Services

The conventions used to define services in clause 8 are those used in ISO/IEC 10728:1993 and where operations on the abstract data structure are defined, they are in terms of the data structure defined in Clause 6 of ISO/IEC 10728.

6 Concepts and facilities

6.1 Overview

This clause describes the Transfer File format and the services which are used to export and import IRDS data.

The services specified provide for the creation of an operating system file of characters of a specified encoding and in a defined sequence and its subsequent use in the same or a different real system. The physical transfer of an Transfer File from one real system to another may take place by any appropriate means.

The character encoding to be used is specified by this International Standard and hence no translation of the Transfer File is required or permitted when the file is transferred from one real system to another.

6.2 Transfer File

6.2.1 Structure

An IRDS Transfer File may contain an IRD Definition (which contains information which defines the content of an IRD) or a subset of an IRD Definition and/or an IRD or a subset of an IRD. For ease of reference each of these combinations is referred to in this part of this International Standard as IRDS-data.

An IRDS Transfer File is made up of three components as follows:

- a) Transfer File Header;
- b) Transfer File IRD Definition;
- c) Transfer File IRD Content.

The Transfer File header is a mandatory component. Both the Transfer File IRD Definition data and the Transfer File IRD Content are optional. However, one of the two must be present. The presence or absence of each is indicated in the Transfer File Header.

There may be many Transfer File IRD Definition Data components and/or Transfer File IRD Content components. This standard requires one component per IRD Working Set.

The reason for the optionality of the IRD Definition Data component is to allow it to be omitted in situations where data conforming to the same IRD Definition is being transferred frequently between the same exporter and importer.

Optionality of the IRD Content component allows an IRD Definition to be exported without any conforming IRD Content. This facility may be used prior to a set of Transfer Files being created using the same IRD Definition, but containing different IRD Content.

The IRD Definition Data and the IRD Content are each composed of a number of sets of rows. Each set of rows represents selected data from a single working set.

All IRD Definition and IRD level tables can be exported. In the data for a working set, all the IRD Object and IRD Object Version rows are included that are referenced by any other rows selected from the working set.

In the data for a working set, rows from the IRD_REFERENCE_PATH table are included to describe all of the reference paths used by foreign keys in the rows selected from the working set.

6.2.2 Transfer File Header

The Transfer File header contains information about the Transfer File, such as date time of file creation and information about the source of the file such as the person responsible for creation. Further user defined information may be added.

6.2.3 Transfer File IRD Definition

The Transfer File IRD Definition data consists of the definition of IRD Content which may then be exported as other components of a Transfer File IRD Content in the same or subsequent Transfer Files.

A Transfer File IRD Definition is for a set of IRD tables that a user of an IRDS-implementation has created and that a user wishing to export IRDS-data has selected.

Each IRD Definition has a name conforming to the rules for names in ISO/IEC 10728. This name is assigned by the exporter of the Transfer File in which the IRD Definition is used.

6.2.4 Transfer File IRD Content component

This component is the IRD Content. The Transfer File IRD Content shall conform (in the IRDS sense) to an IRD Definition which is named in the Transfer File header. The IRD Definition may or may not be present in the same Transfer File.

6.3 Export/import services

6.3.1 *Export services*

There are the following export services:

- a) Open a Transfer File for export;
- b) Select the rows to be exported from specified tables; this service may be invoked many times, to select multiple tables and/or multiple sets of rows from a single table;
- c) Clear a previous selection;
- d) Export the contents of a Working Set; this service may be invoked many times, to export both IRD Definition and IRD level working sets;
- e) to close the export file and make it available to other users.

The Open Export File Service generates the Transfer File header component.

The Export Working Set Service converts a working set in an IRD Definition or an IRD into the format defined for the Transfer File.

6.3.2 *Import services*

There are the following import services:

- a) Open a Transfer File for import;
- b) Import Working Set;
- c) Close IRDS Import File.

Successful invocation of an Import Working Set service applied to an IRD Definition Working Set contained in the Transfer File, creates appropriate rows in the IRD Definition tables in the importing IRDS implementation.

Successful invocation of an Import Working Set service applied to an IRD Working Set contained in the Transfer File, creates appropriate rows in the IRD tables in the importing IRDS implementation.

If the Transfer File contains an IRD Working Set and the Transfer File IRD Definition component is not included, the import service makes use of the name of the IRD Definition given in the Transfer File header component to locate the named IRD Definition in the importing IRDS-implementation.

6.4 Working Sets

Each invocation of the export service generates rows from the single working set that represents the current or specified context. If Full Context is specified, additional rows may be exported from referenced working sets, depending on the selection criteria. The tables from which rows are to be extracted can be selected using the IRDSSelectForExport service.

6.5 Import Control Tables

Tables are defined in clause 8 which can be populated by the IRDS User and by the import services to assist the process of matching objects in an import file with objects already stored in the IRD or IRD Definition.

6.6 Content Modules

Those rows that represent objects contained within one or more IRDS Content Modules may be selected by using the IRDSSelectForExport service and specifying the tables required and a WHERE clause that includes appropriate references to IRD_MODULE_OBJ_KEY and IRD_MODULE_WS_KEY defined in ISO/IEC 10728, 6.1.4.1 Table IRD Object.

7 Specification of the Transfer File

7.1 Character Set

An IRDS Transfer File shall consist of a string of the characters from the Basic Multilingual Plane (BMP) of ISO/IEC 10646-1.

The file header (syntax element <tfIrdHeader> defined in 7.6.2 below) shall contain only characters from Table 1 of ISO/IEC 10646-1.

7.2 Character Coding

Characters in a Transfer File shall be coded using the UCS-2 coding of ISO/IEC 10646.

7.3 Tokens

7.3.1 Terminal Tokens

The characters defined in ISO 9075, 5.1 and indirectly referenced below by reference to clauses in ISO/IEC 9075 are defined to be the respective characters defined in ISO/IEC 10646-1 Table 1.

The following terminal BNF tokens are used in the definition of the Transfer File:

<space> ::=	Defined as the character at row 0 column 002 in ISO/IEC 10646-1.
<carriage return> ::=	Defined as the character at row D column 000 in ISO/IEC 10646-1.
NOTE This conforms to ISO/IEC 6429:1992, 8.3.15.	
<exclamation> ::=	Defined as the character at row 1 column 002 in ISO/IEC 10646-1.

<line feed> ::=	Defined as the character at row A column 000 in ISO/IEC 10646-1.
NOTE This conforms to ISO/IEC 6429:1992, 8.3.75.	
<tab> ::=	Defined as the character at row 9 column 000 in ISO/IEC 10646-1.
NOTE This conforms to ISO/IEC 6429:1992, 8.3.61.	
<double quote> ::= "	Defined as the character at row 2 column 002 in ISO/IEC 10646-1 and referenced in ISO/IEC 9075, 5.3.
<any character except double quote and newline> ::=	any character defined in the basic multilingual plane of ISO/IEC 10646-1 except <double quote> as defined above and <newline> as defined below.
<column name> ::=	as defined in ISO/IEC 9075, 5.4.
<right brace> ::= }	Defined as row D column 007 in ISO/IEC 10646-1 and referenced in ISO/IEC 9075, 5.4.
<left brace> ::= {	Defined as row B column 007 in ISO/IEC 10646-1 and referenced in ISO/IEC 9075, 5.4.
<minus sign> ::= -	Defined as row D column 002 in ISO/IEC 10646-1 and referenced in ISO/IEC 9075, 5.1.
<search condition> ::=	as defined in ISO/IEC 9075, 8.12.
<SQL language identifier> ::=	as defined in ISO/IEC 9075, 5.4.
<table name> ::=	as defined in ISO/IEC 9075, 5.4.
<timestamp literal> ::=	as defined in ISO/IEC 9075, 5.3.
<unsigned integer> ::=	as defined in ISO/IEC 9075, 5.3.

7.3.2 Commonly used tokens

The following BNF tokens are used in a number of places in the definition of the file structure:

<newline> ::= <carriage return> | <line feed>
 <double quote in character string literal> ::= <double quote><double quote>

```

<character string literal> ::=
    <double quote>
        { <character string literal element > } ...
    <double quote>

<character string literal element > ::=
    <any character except double quote and newline> |
    <double quote in character string literal> |
    <newline>

<any character except newline> ::=
    <any character except double quote and newline> |
    <double quote>

<startComment> ::=
    <exclamation><exclamation>

<comment> ::=
    <startComment>
        { <any character except newline> } ...
    <newline>

<separator> ::= { <comment> | <space> | <newline> | <tab> } ...

<endSeq> ::= <right brace>

<startSeq> ::= <left brace>

<quoted SQL language identifier> ::=
    <double quote> <SQL language identifier> <double quote>

```

7.4 White Space

The syntactic token <separator> defines characters that are potentially white space.

Where such characters fall inside < character string literal > as defined above, they are part of the literal. Where such characters fall outside such character string literals they are termed “white space” and are ignored except in so far as they act to separate one token from another.

Rules for including <separator>s in the export/import file are as follows:

1. If two tokens in the BNF definition ARE NOT separated by a space, then <separator>s ARE NOT PERMITTED in the generated result. E.g. <double quote><double quote>.
2. If two tokens in the BNF definition ARE separated by a space, then <separator>s ARE PERMITTED in the generated result.
3. Where the BNF definition includes a constant value (e.g. TABLE_NAME), a <separator> is REQUIRED after this value, even though this is not explicitly shown in the BNF definition.
4. If a <separator> is REQUIRED between two tokens, this is explicitly shown in the syntax definition.

7.5 IRDS Transfer File Structure

7.5.1 Content

The IRDS **Transfer File** contains IRDS-data. It is made up of three components as:

- (1) Transfer File Header;
- (2) Transfer File IRD definition; and
- (3) Transfer File IRD content.

The **Transfer File Header** is the first component of a Transfer File and it contains information that uniquely defines the Transfer File such as the export IRD identification and time stamp. The **Transfer File IRD definition** components are the next optional components of the Transfer File and they contain the IRD definition that defines IRD content that may or may not be included in the same Transfer File. The **Transfer File IRD content** components are the last optional components of the Transfer File and contain the IRD objects that were defined by a previous (in this file or another) Transfer File IRD definition.

7.5.2 Definition

Figure 1 defines the syntax of an <irdsTransferFile>.

```
<irdsTransferFile> ::=  
    <tfIrdHeader> <tfComponent> [ , <tfComponent> ] ...  
<tfComponent> ::=  
    <tfIrdDefinition> | <tfIrdContents>
```

Figure 1 — File Syntax

7.5.3 General Rules

1. <tfIrdHeader> and at least one <tfComponent> must be present.
2. When both <tfIrdDefinition> components and <tfIrdContent> components are present, the <tfIrdDefinition> components shall precede all <tfIrdContent> components.

7.6 Transfer File Header

7.6.1 Content

The Transfer File Header identifies the file as an IRDS Transfer File and is mandatory. The Transfer File Header contains the following groups of properties:

- a) Transfer File Attributes;
- b) Transfer Source Attributes;
- c) Transfer User Parameters.

7.6.2 Definition

Figure 2 defines the syntax of a <tfIrdHeader>.

```

<tfIrdHeader> ::=
    <startSeq>
        <tfAttributes><tfSourceAttributes>
            [<tfUserParameters>]
    <endSeq>
<tfAttributes> ::=
    <startSeq>
        <tfSignature>,
        TITLE <tfTitle>,
        SYNTAX <tfSyntax>,
        ENCODING <tfEncoding>,
        TAGS <tfTableRowTags>,
        CODESET <tfCodeset>,
        TIMESTAMP <tfTimestamp>
    <endSeq>
<tfSignature> ::= DMEI-IRDS
<tfTitle> ::= "ISO/IEC 13238-3:1998 IRDS TRANSFER FILE"
<tfSyntax> ::= BNF
<tfEncoding> ::= CLEAR
<tfTableRowTags> ::= FULL | SHORT | NUMBER | UNLABELLED
<tfCodeset> ::= "ISO/IEC 10646-1 level 1 UCS-2"
<tfTimestamp> ::= <double quote> <timestamp literal> <double quote>
<tfSourceAttributes> ::=
    <startSeq>

```

Figure 2 — Header Syntax (*continued*)

```

SOURCE_NAME      <publisherName> ,
SOURCE_SYSTEM_NAME    <publisherSystemName>

[, SOURCE_FILE_VERSION    <publisherVersion>]
[, SOURCE_ENVIRONMENT    <exporterToolPlatform>]
[, SOURCE_PROGRAM_NAME    <exporterToolName>]
[, SOURCE_PROGRAM_VERSION    <exporterToolVersion>]
[, SOURCE_PROGRAM_VENDOR    <exporterToolVendor>]
<endSeq>

<publisherName> ::=      < character string literal>
<publisherSystemName> ::= < character string literal>
<publisherVersion> ::=   <character string literal>
<exporterToolPlatform> ::= <character string literal>
<exporterToolName> ::=   <character string literal>
<exporterToolVersion> ::= <character string literal>
<exporterToolVendor> ::= <character string literal>
<tfUserParameters> ::=
    , USER_PARAMETERS < character string literal>

```

Figure 2 — Header Syntax (concluded)

7.6.3 General Rules

1. The identifiers shall consist of:

- a) <tfTitle> “ISO/IEC 13238-3:1998 IRDS Transfer File” as a constant.
- b) <tfSyntax>: always BNF. This is the only syntax supported by this International Standard. The field is included to allow for further syntax forms to be prescribed in future versions of this standard or of other standards with a similar file structure.
- c) <tfEncoding>: always CLEAR. This is the only token encoding supported by this International Standard. The field is included to allow for further token encoding forms to be prescribed in future versions of this standard or of other standards with a similar file structure.
- d) <tfCodeset> “ISO/IEC 10646-1 level 1 UCS-2”.

NOTE ISO/IEC 10646-1 defines a number of conformance levels, of which level 1 is prescribed for an IRDS Transfer File. ISO/IEC 10646-1 defines a number of forms of a coded character set, of which UCS-2 is prescribed for an IRDS Transfer File.

- e) <tfTableRowTags> One of FULL, SHORT, NUMBER OR UNLABELLED constants. FULL means that the actual column names are used. SHORT means that the first three characters of each word in the column name, excluding the “IRD” prefix are used as column identifiers. NUMBER means that the column position is used as the tag. UNLABELLED means that the column values are not identified, and are placed in the file in the order of their definition.

- f) <tfTimestamp> provides the date and time that the file was created in the format of an SQL <timestamp literal> expressed as UTC.
2. The other parameters are implementation or user defined:
- a) <publisherName> is used to identify the person or group responsible for the preparation of the Transfer File. It is supplied on the IrdsOpenExportFile call.
 - b) <PublisherSystemName>: is a globally unique identifier assigned to the publisher's system, such as an OSI Directory name, a DNS name, or an IP Address. It is supplied on the IrdsOpenExportFile call.
 - c) <publisherVersion>: is a version number assigned to the file by its publisher. It is supplied on the IrdsOpenExportFile call.
 - d) <exporterToolPlatform> identifies the hardware/system software platform of the tool used by the publisher.
 - e) <exporterToolName> is a unique identifier for the exporting tool or product.
 - f) <exporterToolVersion> enables the exporter name to be qualified with a version identifier.
 - g) <exporterToolVendor> identifies the provider of the tool used by the publisher.
3. <tfUserParameters>, the user parameters, are optional. If required they are supplied on the IrdsOpenExportFile call.

7.7 Transfer File IRD definition

7.7.1 Content

The IRD Definition component serves to define the data that may follow in the IRD Content component of the same or a subsequent Transfer File. The IRD Definition component is made of the following parts:

- a) a component header part;
- b) a component content part.

The component content part contains tables.

7.7.2 Definition

Figure 3 defines the syntax of a <tfIrdDefinition>.

```

<tfIrdDefinition> ::=
    <startSeq>
        >tfIrdDefinitionHeader>,
        <irdDefWorkingSetExport> [, <irdDefWorkingSetExport>] ...
    <endSeq>

<tfIrdDefinitionHeader> ::=
    COMPONENT_TYPE <tfComponentType>,
    COMPONENT_TITLE <tfIrdDefTitle>,
    COMPONENT_NAME <irdDefName>

<tfComponentType> ::= DMEI-IRDS-DEFINITION

<tfIrdDefTitle> ::= "ISO/IEC 10728:1993 IRDS SERVICES INTERFACE IRD
DEFINITION"

NOTE A later version of the International Standard may be referenced.

<irdDefName> ::= <quoted SQL language identifier>.

<irdDefWorkingSetExport> ::=
    <startSeq>
        <workingSetTableRow>,
        <tfTable> [, <tfTable>] ...
    <endSeq>

```

Figure 3 — IRD Definition Export Syntax

7.7.3 General Rules

1. The <irdDefWorkingSetExport> contains data extracted from a single working set. If more than one working set needs to be exported, then multiple <irdDefWorkingSetExport> components are used.
2. Any Time or Timestamp data types within the exported data shall be expressed in UTC.
3. All data are exported from the abstract data structures defined in Chapter 6 of ISO/IEC 10728 IRDS Services Interface and are not reproduced in this International Standard. While any table may be exported, the most relevant tables are IRD specific as listed in the services interface subclause 5.3. The list in the following table is provided as an example for reference only by naming Table 8 through 24 as a typical requirement. The export user may select any set of IRD tables that are considered to be relevant.

#	Code	Table Name
08	SGR	IRD_SCHEMA_GROUP
09	SCH	IRD_SCHEMA
10	SRE	IRD_SCHEMA_REFERENCE
11	DTD	IRD_DATA_TYPE_DESCRIPTOR
12	DOM	IRD_DOMAIN
13	TAB	IRD_TABLE
14	VIE	IRD_VIEW
15	COL	IRD_COLUMN
16	VTU	IRD_VIEW_TABLE_USAGE
17	VCU	IRD_VIEW_COLUMN_USAGE
18	TCO	IRD_TABLE_CONSTRAINT
19	KCU	IRD_KEY_COLUMN_USAGE
20	RCO	IRD_REF_CONSTRAINT
21	CCO	IRD_CHECK_CONSTRAINT
22	CTU	IRD_CHECK_TABLE_USAGE
23	CCU	IRD_CHECK_COLUMN_USAGE
24	ASS	IRD_ASSERTION

Figure 4 — IRD Tables

7.8 Transfer File IRD content

7.8.1 Content

The dictionary contents are held in the third component as the Transfer File data. IRD content to be exported will be in the context of a current user session.

The IRD content component is made of the following parts:

- a) a component header part;
- b) a component content part.

The component content part contains tables.

7.8.2 Definition

Figure 5 defines the syntax of a <tfIrdContents>.

```

<tfIrdContents> ::=
    <startSeq>
        <tfIrdContentsHeader>,
        <irdWorkingSetExport> [, <irdWorkingSetExport>] ...
    <endSeq>

<tfIrdContentsHeader> ::=
    COMPONENT_TYPE <tfComponentType>,
    COMPONENT_TITLE <tfIrdTitle>,
    COMPONENT_NAME <irdDicName>,
    COMPONENT_DEFINITION_NAME <irdDefName>,
    COMPONENT_SCHEMA_GROUP_NAME
        <irdSchemaGroupName>

<tfComponentType> := DMEI-IRDS-CONTENT

<tfIrdTitle> ::=
    "ISO/IEC 10728:1993 IRDS SERVICES INTERFACE IRD CONTENT"

NOTE A later version of the International Standard may be referenced.

<irdDicName> ::= <quoted SQL language identifier>

<irdDefName> ::= <quoted SQL language identifier>.

<irdSchemaGroupName> ::= <quoted SQL language identifier>

<irdWorkingSetExport> ::=
    <startSeq>
        <working SetTableRow>,
        <tfTable>[, <tfTable>] ...
    <endSeq>

```

Figure 5 — IRD Content Syntax

7.8.3 General Rules

1. The <irdWorkingSetExport> contains data extracted from a single working set. If more than one working set needs to be exported, then multiple <irdWorkingSetExport> components are used.
2. Any Time or Timestamp data types within the exported data shall be expressed in UTC.

7.9 Transfer File Table Transfer Syntax

7.9.1 Content

IRD definition and dictionary content are transferred using the same tabular structure, referred to as <tfTable>

The Table Transfer Structure is made of the following parts:

- a) a table header part;
- b) a table rows part.

The table rows part contains column values for each row.

7.9.2 Definition

Figure 6 defines the syntax of a <tfTable>.

```

<tfTable> ::=
    <startSeq>
        <tfTableHeader>,
        <tfRow> [, <tfRow>] ...
    <endSeq>

<tfTableHeader> ::=
    TABLE_NAME <tfTableName>
    [, TAGS <tfTableRowTags> ]

<tfTableName> ::= " <table name> "

<tfTableRowTags> ::= FULL | SHORT | NUMBER | UNLABELLED

<tfRow> ::=
    <startSeq>
        <tfColumn> [, <tfColumn>] ...
    <endSeq>
  
```

Figure 6 — Table Transfer Syntax (*continued*)

```

<tfColumn> ::= [ <tfColumnId> <separator> ] <tfColumnValue>
<tfColumnId> ::= <tfColumnName> | <tfColumnTag> | <tfColumnNumber>
<tfColumnName> ::= <double quote> <column name> <double quote>
<tfColumnTag> ::= <SQL language identifier>
<tfColumnNumber> ::= <unsigned integer>
<tfColumnValue> ::= <character string literal>
<workingSetTableRow> ::=
    <startSeq>
        TABLE_NAME IRD_WORKING_SET,
        [ TAGS <tfTableRowTags> ]
        <tfRow>
        !! for a row from the IRD Working Set Table
    <endSeq>

```

Figure 6 — Table Transfer Syntax (concluded)

7.9.3 General Rules

1. If the value of a column in a row is null, then no instance of <tfColumn> is to be written to the file.
2. <tfTableRowTags> shall be one of FULL, SHORT, NUMBER or UNLABELLED constants. FULL means that the actual column names are used. SHORT means that the first three characters of each word in the column name, excluding the “IRD” prefix are used as column identifiers. NUMBER means that the column position is used as the tag. UNLABELLED means that the column values are not identified, and are placed in the file in the order of their definition.
3. If <tfTableRowTags> = FULL then


```

<tfColumn> ::= <tfColumnId><separator><tfColumnValue>
<tfColumnId> ::= <tfColumnName>

```
4. If <tfTableRowTags> = SHORT then


```

<tfColumn> ::= <tfColumnId><separator><tfColumnValue>
<tfColumnId> ::= <tfColumnTag>

```

5. If `<tfTableRowTags>` = NUMBER then

`<tfColumn>` ::= `<tfColumnId><separator><tfColumnValue>`

`<tfColumnId>` ::= `<tfColumnNumber>`

6. If `<tfTableRowTags>` = UNLABELLED then

`<tfColumn>` ::= `<tfColumnValue>`

7. If `<tfTableRowTags>` is omitted in `<tfTableHeader>`, the value from `<tfIrdHeader>` is used.

If `<tfTableRowTags>` is present in `<tfTableHeader>`, its value overrides, for this table, the value specified in `<tfIrdHeader>`.

8 Specification of the Import Control Tables

The tables specified in this clause are an extension of the IRD Definition specified in ISO/IEC 10728.

8.1 Table IRDS Dictionary Import Control

8.1.1 *Function*

This table has one row for each external Dictionary for which Import Control data is to be maintained. The rows in the table are created by the `IrdsImportWorkingSet` service, based on information in the `<tfIrdHeader>`, and/or they may be maintained by an IRDS User to control future imports.

8.1.2 *Definition*

CREATE TABLE IRDS_DICTIONARY_IMPORT_CONTROL

```
(
  IRDS_EXT_DICT_OBJ_KEY IRDS_KEY NOT NULL,
  IRDS_EXT_DICT_WS_KEY IRDS_KEY NOT NULL,
  PRIMARY KEY (IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY),
  IRDS_EXT_SYSTEM_NAME CHAR(255) NOT NULL,
  IRDS_EXT_DICT_NAME CHAR(255) NOT NULL,
  UNIQUE (IRDS_EXT_SYSTEM_NAME, IRDS_EXT_DICT_NAME)
)
```

8.1.3 *Description*

1. `IRDS_EXT_DICT_OBJ_KEY` and `IRDS_EXT_DICT_WS_KEY` together form the primary key of the IRD Dictionary Import Control Table.
2. `IRDS_EXT_DICT_NAME` identifies the Dictionary within the system named `IRDS_EXT_SYSTEM_NAME`, from which a Transfer File was created, whose data is to be imported to the local IRDS.
3. `IRDS_EXT_SYSTEM_NAME` and `IRDS_EXT_DICT_NAME` uniquely identify the Dictionary.
4. The size of `IRDS_EXT_SYSTEM_NAME` and of `IRDS_EXT_DICT_NAME` is not constrained by the local implementation limit on IRDS Names, because the exporting system may have different values.

8.2 Table IRD Object Import Control

8.2.1 *Function*

This table has one row for each IRD Object to be mapped from an external dictionary to a local dictionary. The rows in the table are created by the `IrdsImportWorkingSet` service, based on the results of object matching, and/or they may be maintained by an IRDS User to control future imports.

8.2.2 Definition

CREATE TABLE IRD_OBJECT_IMPORT_CONTROL

```
(
  IRDS_EXT_DICT_OBJ_KEY IRDS_KEY NOT NULL,
  IRDS_EXT_DICT_WS_KEY IRDS_KEY NOT NULL,
  IRD_IMP_CONT_OBJ_KEY IRDS_KEY NOT NULL,
  PRIMARY KEY (IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY,
    IRDS_IMP_CONT_OBJ_KEY),
  CONSTRAINT IMPORTED_OBJECT_COMES_FROM_EXTERNAL_DICTIONARY
    FOREIGN KEY (IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY)
      REFERENCES IRDS_DICTIONARY_IMPORT_CONTROL
        ON DELETE CASCADE,
  IRD_OBJECT_KEY IRDS_KEY,
  CONSTRAINT IMPORTED_OBJECT_MAPS_TO_IRD_OBJECT
    FOREIGN KEY (IRD_OBJECT_KEY)
      REFERENCES IRD_OBJECT
        ON DELETE CASCADE
)
```

8.2.3 Description

1. IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY and IRDS_IMP_CONT_OBJ_KEY together form the primary key of the Ird Object Import Control table.
2. IRDS_IMP_CONT_OBJ_KEY is the key of an object in the Transfer File that is to be mapped to an object in the local IRDS.
3. IRD_OBJECT_KEY is the key of an object (a row in IRD_OBJECT table) in the local IRDS to which the object in the Transfer File is mapped.

8.3 Table IRD Working Set Import Control**8.3.1 Function**

This table has one row for each IRD Working Set that has been imported from an E/I file to a local working set. The rows in the table are created by the IrdImportWorkingSet service, based on the parameters provided to the service, and/or they may be maintained by an IRDS User to control future imports.

8.3.2 Definition

CREATE TABLE IRD_WORKING_SET_IMPORT_CONTROL

```
(
  IRDS_EXT_DICT_OBJ_KEY IRDS_KEY NOT NULL,
  IRDS_EXT_DICT_WS_KEY IRDS_KEY NOT NULL,
  IRD_IMP_CONT_WS_KEY IRDS_KEY NOT NULL,
  IRD_WORKING_SET_KEY IRDS_KEY NOT NULL,
  PRIMARY KEY (IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY,
    IRDS_IMP_CONT_WS_KEY, IRD_WORKING_SET_KEY),
  CONSTRAINT IMPORTED_WS_COMES_FROM_EXTERNAL_DICTIONARY
    FOREIGN KEY (IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY)
      REFERENCES IRDS_DICTIONARY_IMPORT_CONTROL
        ON DELETE CASCADE,
  CONSTRAINT IMPORTED_WS_MAPS_TO_IRD_WORKING_SET
    FOREIGN KEY (IRD_WORKING_SET_KEY)
      REFERENCES IRD_WORKING_SET
        ON DELETE CASCADE
)
```

8.3.3 Description

1. IRDS_EXT_DICT_OBJ_KEY, IRDS_EXT_DICT_WS_KEY and IRDS_IMP_CONT_WS_KEY and IRD_WORKING_SET_KEY together form the primary key of the Ird Working Set Import Control table.
2. IRDS_IMP_CONT_WS_KEY represents the WS_KEY part of primary and foreign keys in the content of an Transfer File, that is to be mapped to a working set in the local IRDS.
3. IRD_WORKING_SET_KEY is the key of a row in the IRD_WORKING_SET table in the local IRDS, to which the WS_KEY in the Transfer File is mapped.
4. When the IrdImportWorkingSet service is invoked, a row will be created (if it does not already exist) mapping the working set to be imported, to the local working set into which the data is to be imported. Ird users may insert other rows in this table to allow foreign key references in one working set to be mapped to another in the local IRDS.

9 Specification of the export/import services

9.1 Relationship to the IRDS Services Interface

The services defined in this clause are additional to those defined in ISO/IEC 10728 Services Interface under subclause 9.2 Level Independent Services. In this Part of 13238 Pascal Language Bindings are given.

9.2 Services provided

Services are provided to:

- a) Open the Transfer File for Export;
- b) Select the rows to be exported from specified tables;
- c) Clear a previous selection;
- d) Export a single working set or a related set of working sets;
- e) Close the Export File;
- f) Open the Transfer File for Import;
- g) Import a single Working Set;
- h) Import the entire Transfer File;
- i) Close the Import File.

9.3 Selection of data to be exported

Selection takes place within the context of an IRDS Session.

When not all tables are to be exported, the required tables are specified by invoking the IRDSSelectForExport service prior to invoking the IRDSExportWorkingSet service. This service also allows specific rows to be selected.

Each invocation of an IRDSSelectForExport service selects a set of rows from a named table.

The IRDSSelectForExport service may be invoked many times. The effect of multiple calls is to define multiple sets of rows for export by the IRDSExportWorkingSet service. Multiple IRDSSelectForExport service calls may name the same table. The effect is to select several sets of rows from the same table. Although a row may thus be selected several times, a selected row is exported exactly once.

A previously established set of selection criteria may be cancelled by invoking the IRDSClearExportSelection service.

9.4 Service Data Types and Data Structures

This subclause defines data types and data structures used in the definition of the export/import services. The following are defined in ISO/IEC 10728:

IrdsRetCode

The following are defined in this clause:

```
type IrdsTFileHandle = integer;
type MissingTablesController = (
    kCreateMissingTables,
    kSkipMissingTables,
    kFailOnMissingTable);
type SkipFailingRowsController = (
    kSkipFailingRows,
    kDontSkipFailingRows);
type SqlCommitController = (
    kCommitAtEndOfFile,
    kCommitAfterEachTable,
    kCommitAfterEachTableAndEveryNRows);
type UpdateExistingRowsController = (
    kUpdateExistingRows,
    kDontUpdateExistingRows);
```

9.5 Open IRDS Export File

9.5.1 *Function*

Invocation of this service causes an output file to be opened and the header defined by <tfIrdHeader> to be written to the file.

9.5.2 *Format*

procedure IrdsOpenExportFile

```
( ExpFilePath:      String;
  ExpFileName:      String;
  UserParm:         IrdsText;
  Tags:             String;
  PublisherName:     String;
  PublisherSystemName: String;
  PublisherVersion:  String;
  var CurrTFileHandle: IrdsTFileHandle;
  var RetCode:       IrdsRetCode );
```

9.5.3 *Input*

The following variables may be set:

ExpFilePath := when not null, the path to the directory in which the file is to be placed. When null, a default path will be used.

ExpFileName := when not null, the name to be used for the export file. When null, a default file name will be used.

UserParm = when not null, the value is user defined to add extra user information to the TF header.

Tags = one of the values defined for <tfTableRowTags> in 7.9.2.

PublisherName= the value to be included in the header as <publisherName>.

PublisherSystemName= the value to be included in the header as <PublisherSystemName>.

PublisherVersion= the value to be included in the header as <publisherVersion>

9.5.4 *Output*

The following variables shall be set:

CurrTFileHandle := a system handle for the opened export file.

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
92	901: The user does not have the appropriate operating system file access privileges.
92	902: Invalid File Name
92	903: Specified File Path does not exist
92	904: Other Operating System File Error
92	905: Invalid TAGS parameter.

9.5.5 General Rules

1. More than one IRDS Transfer File may be open at any one time. Use of the CurrTFileHandle allows the IRDS to relate specific invocations of IrdsSelectForExport, IrdsClearExportSelection, IrdsExportWorkingSet and IrdsCloseExportFile to a specific Transfer File.

2. If the file identified by the combination of “ExpFilePath” and “ExpFileName” does not exist, it is created. If the file already existed, its contents are replaced by the data exported by the series of “IrdsExportWorkingSet” services between an IrdsOpenExportFile and the following IrdsCloseExportFile.

3. An instance of <tfIrdHeader> is written to the file. Constant fields are given the values specified in clause 7. The variable fields are completed as follows:

- a) <tfTimestamp> to be a time between the start and end of the invocation of the IrdsOpenExportFile service.
- b) <exporterToolName> an implementor defined name of the tool generating the Transfer File.
- c) <exporterToolVersion> an implementor defined version name of the tool generating the Transfer File.
- d) <publisherName> the UserID of the IRDS User from the IRDSOpen service invocation that established the current session.
- e) <tfUserParameters> the value of the UserParm parameter supplied to the IrdsOpenExportFile call.

9.5.6 Operations on the Abstract Data Structure

None.

9.6 Select For Export Service

9.6.1 Function

This service identifies selection criteria for a set of rows from a named IRD Definition or IRD Level table, for use in a subsequent Export Working Set service. The selection criteria are specified in the form of a SQL <search condition>;

9.6.2 Format

```
procedure IRDSSelectForExport (
    CurrTFileHandle:      IrdsTFileHandle;
    Tablename:            IrdsName;
    SelectionExpression :  String;
    var RetCode:          IrdsRetCode );
```

9.6.3 Input

The following variables are to be set:

CurrTFileHandle := the handle of the Transfer File previously opened by a successful invocation of IrdsOpenExportFile and not yet the subject of an invocation of IrdsCloseExportFile.

Tablename := the name of an IRD or IRD Definition Level table whose rows are to be selected for export.

SelectionExpression := an SQL <search condition> for use in a subsequent Export Working Set service to select rows that conform to the <search condition>.

9.6.4 Output

The following variable shall be set:

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
07	000: Invalid column name
02	008: Table does not exist
25	004: Invalid Transfer File handle
42	000: User does not have the appropriate privileges
10	000: Syntax error in <search condition>

9.6.5 General Rules

1. Tablename shall identify an IRD or IRD Definition Level table.
2. SelectionExpression shall be a valid <search condition> as defined in ISO/IEC 9075.
3. The <search condition> shall be validated against the specified table for the current user.
4. The IRDSSelectForExport function may be invoked many times before the IrdsExport is called.
5. The accumulated selection criteria remain in effect until the invocation of a Clear Export Selection service or a Close IRDS service for the specified IRDS session.

9.6.6 Operations on the Abstract Data Structure

None.

9.7 Clear Export Selection Service

9.7.1 *Function*

This service clears any previously established selections for export to the nominated Transfer File.

9.7.2 *Format*

```
procedure IRDSClearExportSelection (
    CurrTFileHandle:   IrdsTFileHandle;
    var RetCode:       IrdsRetCode );
```

9.7.3 *Input*

The following variable is to be set:

CurrTFileHandle:= the handle of the Transfer File previously opened by a successful invocation of IrdsOpenExportFile and not yet the subject of an invocation of IrdsCloseExportFile.

9.7.4 *Output*

The following variable shall be set:

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
25	004: Invalid Transfer File handle

9.7.5 *General Rules*

None.

9.7.6 *Operations on the Abstract Data Structure*

None.

9.8 Export Working Set

9.8.1 Function

This service exports data from a working set to the nominated Transfer File. The working set identification functionality of this service is closely related to that of the Open Cursor service.

NOTE Working Sets from more than one IRD or IRD Definition may be exported to the same Transfer File.

9.8.2 Format

procedure IrdsExportWorkingSet

```
( CurrTFileHandle:  IrdsTFileHandle;  
  CurrSessId:      IrdsSessId;  
  UseContext:      boolean;  
  WkgSetName:      IrdsName;  
  WkgSetVerId:     IrdsName;  
  FullContext:     boolean;  
  ExpDef:          boolean;  
  var RetCode: IrdsRetCode );
```

9.8.3 Input

The following variables may be set:

CurrTFileHandle := the handle of the Transfer File previously opened by a successful invocation of IrdsOpenExportFile and not yet the subject of an invocation of IrdsCloseExportFile.

CurrSessId := the current session identifier returned by a previous Open IRDS service.

UseContext := when true, indicates that only the object versions in the current context are to be exported and the values of “WkgSetName” and “WkgSetVerId” are to be ignored; when false, indicates that the variables “WkgSetName” and “WkgSetVerId” are to be used in establishing the context of the scope of export.

WkgSetName := when “UseContext” is false, provides the name of the working set to be used as the context of the export.

WkgSetVerId := when “UseContext” is false, provides the version identifier of the working set to be used as the context of the export.

FullContext := when true, a full context is to be used; when false, a single context is to be used.

ExpDef := when true, then the export definition is to be exported. When false, then the export definition is not exported and the IRD names are used to identify the definition to the importer.

9.8.4 Output

The following variables shall be set:

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
02	003: Working set does not exist
03	001: No data selected
25	001: Invalid Session identifier
25	004: Invalid Transfer File handle
25	007: Attempt to export IRD Definition Level Working Set after exporting an IRD Level Working Set to the same file
42	000: User does not have the appropriate privileges
54	000: No current working set context has been established

9.8.5 General Rules

1. A successful call to `IrdsOpen` shall be made prior to calling `IrdsExportWorkingSet` and the session id returned by the `IrdsOpen` call shall be passed to the `IrdsExportWorkingSet` call.
2. A successful call to `IrdsOpenExportFile` shall be made prior to calling `IrdsExportWorkingSet` and the `CurrTFileHandle` returned by the `IrdsOpenExportFile` shall be passed to the `IrdsExportWorkingSet` call.
3. The first call to `IrdsExportWorkingSet` for a particular `IrdsSessId` generates a `<tfIrdDefinitionHeader>` if the current session is an IRD Definition Level session or a `<tfIrdContentsHeader>` component if the current session is an IRD Level session.
4. `<irdDefName>` is to be taken from the `irdDefName` parameter of the `IrdsOpen` service invocation that established the current IRDS session.
5. `<irdDicName>` is to be taken from the `IrdDicName` parameter of the `IrdsOpen` service invocation that established the current IRDS session if the session is an IRD Level session.
6. Invocation of this service generates one or more entries in the Transfer File with the structure defined by either `<irdDefWorkingSetExport>`, if the current session is an IRD Definition Level session or by `<irdWorkingSetExport>` if the current session is an IRD Level session.
7. When `Full Context` is true an `<irdDefWorkingSetExport>` or a `<irdWorkingSetExport>` is generated for each of the working sets in the full context for which one or more rows have been selected directly or by reference from another selected row.
8. When `"UseContext"` is false, `"WkgSetName"` and `"WkgSetVerId"` shall together specify a working set and this shall be used as the context specifier for the export;
when `"UseContext"` is true, there shall already be a current working set context and this shall be used as the context specifier for the export. When `"FullContext"` is true, the full context defined by the context specifier shall be used; when `"FullContext"` is false, the single context shall be used.

9. If no calls to IRDSSelectForExport have been made, all objects within the specified context are exported. If calls to IRDSSelectForExport have been made, all selected objects within the specified context are exported.

10. A single instance of <workingSetRow> containing the column values for the selected working set, followed by instances of <tfTable> are written to the Export File until a complete <irdDefWorkingSetExport> or <irdWorkingSetExport> has been written. Within each <tfTable> the name of the table being exported is written to the <tfTableName>. For every selected row in the table a <tfRow> is written to the export file. A row is selected if either: (i) no Select For Export Services has been invoked since the previous Clear Export Selection or IRDS Export Working Set Service or (ii) the row meets the selection condition of any Select For Export Service invoked since the previous Clear Export Selection or IRDS Export Working Set Service.

11. If ExpDef is true, an instance of <tfIrdDefinition> is written to the file.

12. The IRDS User who invokes this service shall have a select privilege on the current working set when UseContext is true, or on the specified working set when UseContext is false.

13. Information on implementor limits may be exported for the benefit of the importer, but the implementor limits themselves can not be imported, since the importing implementation's limits are set by the importing implementation itself.

14. Usage Tables (those with the word "usage" as part of their names) may be exported for the benefit of the importer, but are not to be imported and their rows are to be recreated when the appropriate check or view column values are imported.

15. Successive calls to IrdExportWorkingSet for a particular IrdSessId generate new instances of <irdDefWorkingSetExport> or <irdWorkingSetExport> as appropriate within the same <tfComponent>. A call to IrdExportWorkingSet for a different IrdSessId initiates a new <tfComponent>.

16. For a particular Transfer File, all invocations of this service for IRD Definition Level sessions must precede all invocations of this service for IRD Level sessions.

17. Once an IrdExportWorkingSet service has been invoked, committed updates by other users or by other sessions of the invoking user shall not be visible until the service invocation completes. Thus, what is exported is a snapshot of the working set at the time that the service invocation starts.

NOTE No facilities for maintaining consistency across multiple invocations of IrdExportWorkingSet are specified in this International Standard.

9.8.6 Operations on the Abstract Data Structure

None.

9.9 Close IRDS Export File

9.9.1 Function

Invocation of this service causes the output file to be closed and the file made accessible by other users.

9.9.2 Format

```
procedure IrdsCloseExportFile
  ( CurrTFileHandle:  IrdsTFileHandle;
    var RetCode: IrdsRetCode );
```

9.9.3 Input

The following variables shall be set:

CurrTFileHandle := the handle of the Transfer File previously opened by a successful invocation of IrdsOpenExportFile and not yet the subject of a previous invocation of IrdsCloseExportFile.

9.9.4 Output

The following variables shall be set:

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
25	004: Invalid Transfer File handle
25	005: No successful prior call to IrdsOpenExportFile
92	904: Other Operating System File Error

9.9.5 General Rules

1. A successful call to IrdsOpenExportFile shall be made prior to calling IrdsCloseExportFile, and the CurrTFileHandle returned by the IrdsOpenExportFile shall be passed to the IrdsCloseExportFile call.

9.9.6 Operations on the Abstract Data Structure

None.

9.10 Open IRDS Import File

9.10.1 Function

Invocation of this service causes an input file to be opened, the header defined by <tfIrdHeader> to be read from the file and its contents returned.

9.10.2 Format

```
procedure IrdOpenImportFile
( ImpFilePath:   String;
  ImpFileName:   String;
  UpdateExistingRowsIndicator: UpdateExistingRowsController;
  SkipFailingRowsIndicator: SkipFailingRowsController;
  MissingTablesControlIndicator: MissingTablesController;
  ErrorFilePath: String;
  ErrorFileName: String;
  CommitControlIndicator: SqlCommitController;
  CommitAfter: Integer;
  var CurrTFileHandle: IrdstFileHandle;
  var UserParm: IrdstTxt;
  var RetCode: IrdstRetCode );
```

9.10.3 Input

The following variables may be set:

ImpFilePath := when not null, the specified import path will be used. When null, a default import path will be used.

ImpFileName := when not null, the specified file name will be used, in conjunction with the specified path. When null, a default file name will be used.

UpdateExistingRowsIndicator := An indicator that controls whether existing rows are to be updated or not.

MissingTablesControlIndicator := An indicator that controls the action to be taken when a table is found in the import file that is not in the importing IRD, IRD Definition or was not included in the IRD Definition being imported. Allowable actions are:

- a) to create the missing table (kCreateMissingTable);
- b) to halt the import (kFailOnMissingTable);
- c) to skip the table in error (kSkipMissingTables).

SkipFailingRowsIndicator := An indicator that controls whether when a row that fails constraints is encountered, the import halts (kDontSkipFailingRows) or is skipped (kSkipFailingRows).

ErrorFilePath := The path for the directory, in which the error file will be created. If null an implementor defined default path will be used.

ErrorFileName:= The name of the error file that will be created. If null an implementor defined default name will be used.

CommitControlIndicator:= An indicator that controls how often updates should be committed during the import.

CommitAfter:= The number of rows after which a commit will take place if CommitControlIndicator is set to kCommitAfterEachTableAndEveryNRows.

9.10.4 Output

The following variables shall be set:

CurrTFFileHandle := a system handle for the opened import file.

UserParm := the value of the UserParameter in the TF Header; null if no such parameter is present.

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
25	004: Invalid Transfer File Handle
92	901: The user does not have the appropriate operating file system access privileges
92	902: Invalid File Name
92	903: Specified File Path does not exist
92	904: Other Operating System File Error

9.10.5 General Rules

1. Only one IRDS Import file may be open at one time within a specific IRDS Session.
2. If the file identified by the combination of "ImpFilePath" and "ImpFileName" does not exist, an error is raised.

9.10.6 Operations on the Abstract Data Structure

None.

9.11 Retrieve Import File Working Set Details

9.11.1 Function

Each invocation of the `IrdsRetrieveImportFileWorkingSetDetails` service retrieves the details of a Working Set in the currently open IRDS Import file.

9.11.2 Format

```
procedure IrdsRetrieveImportFileWorkingSetDetails
(
    CurrTFileHandle:    IrdsTFileHandle;
    ImpWSCols:          IrdsColList;
    var RetCode:        IrdsRetCode
);
```

9.11.3 Input

The following variables shall be set:

CurrTFileHandle := the system handle for the import file returned by a previous `IrdsOpenImportFile` call.

9.11.4 Output

The following variables shall be set:

RetCode

Possible error states are:

<u>State Class</u>	<u>Subclass</u>
03	002: No further data (No more Working Sets in Import File).
23	005: Syntax error in import file.
25	004: Invalid Transfer File Handle.
25	006: No IRDS Import File open.

ImpWSCols

Each invocation of the service returns the values of the columns of a Working Set written to the Transfer File as an instance of `<WorkingSetTableRow>`.

9.11.5 General Rules

1. The first invocation of the `IrdsRetrieveImportFileWorkingSetDetails` service following a successful invocation of the `IrdsOpenImportFile` service returns the columns describing the first working set in the file in `ImpWSCols`.
2. Each successive call returns the details of the next Working Set in the file until the end of the file is reached when error code 03 subclass 002 is returned.

9.12 Import Working Set

9.12.1 Function

The Import Working Set service shall copy data contained in a single <irdDefWorkingSetExport> or <irdWorkingSetExport> from the Transfer File to a Working Set in the target IRDS. The working set may already exist or may be created by the service.

9.12.2 Format

procedure IrdsImportWorkingSet

```
( CurrSessId:      IrdsSessId;
  CurrTFileHandle: IrdsTFileHandle;
  SearchWkgSetName: IrdsName;
  SearchWkgSetVerId: IrdsName;
  RepContext:      boolean;
  NewWkgSetName:   IrdsName;
  NewWkgSetVerId:  IrdsName;
  NewDcs:          IrdsName;
  BasisWkgSetName: IrdsName;
  BasisWkgSetVerId: IrdsName;
  UseImportedVersionability: boolean;
  Versionable:     boolean;
  SetTimestamp:    boolean;
  SetUser:         boolean;
  var RetCode:     IrdsRetCode );
```

9.12.3 Input

The following variables shall be set:

CurrSessId := the current session identifier returned by a previous Open IRDS service.

CurrTFileHandle := the system handle for the import file returned by a previous IrdsOpenImportFile call.

RepContext := when true, indicates that imported objects with the same unique identifier as those already in the same working set version are to be replaced; when false, indicates that the imported objects that match existing objects in the same working set version are to be rejected.

SetTimestamp := if True, use the imported timestamp; if False, use the current date and time.

SetUser := if True, use the imported User; if False, use the identification of the importing User.

UseImportedVersionability := if True the value of Versionable for the Working Set being imported is used if a new working set is created by this service, if False the value of Versionable is used.

Versionable := the value to be used for the Versionable Column (See ISO/IEC 10728, 6.1.4.2) if a new working set is created by this service.

The following variables may be set:

SearchWkgSetName := the working set name of the working set to be searched for in the Import file. If SearchWkgSetVerId is not null then SearchWkgSetName shall not be null.

SearchWkgSetVerId := the working set version ID of the working set to be searched for in the Import file.

NewWkgSetName := when not null, provides the name of a new working set to be used as the context of the import; when null, the current working set is used as import context. If NewWkgSetVerId is not null then NewWkgSetName shall not be null.

NewWkgSetVerId := when not null, provides the version identifier of a new working set to be used as the context of the import; when null, see General Rules 6 & 7 below.

NewDcs := the initial IRD content status of the new working set.

BasisWkgSetName := when not null, provides the name of a working set on which the new working set is to be based; when null, the new working set is to be initially empty. If “BasisWkgSetVerId” is not null then “BasisWkgSetName” shall not be null.

BasisWkgSetVerId := when not null, provides the version identifier of a working set on which the new working set version is to be based; when null the current working set version is used. If “BasisWkgSetName” is not null then “BasisWkgSetVerId” shall not be null.

9.12.4 Output

The following variables shall be set:

RetCode := with the following error states:

<u>State Class</u>	<u>Subclass</u>
02	003: Working set does not exist (applies to the basis working set.)
03	001: No data selected
02	008: Table does not exist
03	003: Search Working Set does not exist in import file
22	All subclasses are applicable.
23	003: Row failed constraint(s)
23	004: Row in file does not match definition for table
23	005: Table in Import file is not in existing dictionary and MissingTablesControlIndicator is kFailOnMissingTable.
23	006: Basis Working Set specified for existing Working Set
25	001: Invalid session identifier