CDS/ISIS for Windows

Reference Manual

(Version 1.5)

UNESCO
Information Society Division
Sector of Communication and Information

© UNESCO, June 2004

(This manual refers to Winisis 1.5 build 3)

2nd revision by: Ben Winnubst, New Zealand (May 2008)

Foreword

CDS/ISIS is a menu-driven generalized Information Storage and Retrieval system designed specifically for the computerized management of structured non-numerical data bases. One of the major advantages offered by the generalized design of the system is that CDS/ISIS is able to manipulate an unlimited number of data bases each of which may consist of completely different data elements. Although some features of CDS/ISIS require knowledge of and experience with computerized information systems, once an application has been designed the system may be used by persons having had little or no prior computer experience. For advanced users, CDS/ISIS offers a wide range of programming facility allowing the development of specialized applications through the use of its powerful print formats. For real computer programmers, an external programming library, the ISIS_DLL[1], provide all necessary tools for developing CDS/ISIS based applications.

The first version of this manual, referring to CDS/ISIS 1.3 for Windows, was written by the creator of the original CDS/ISIS, Giampaolo Del Bigio. It describes the operations of the Windows version of CDS/ISIS and is meant to be **complementary** to other CDS/ISIS manuals, such as the *CDS/ISIS for Windows Handbook* and the *CDS/ISIS Reference Manual* for the MS-DOS version. In particular, it describes changes and/or new features which are only available in the Windows version. This version is fully compatible with the MS-DOS[2] version of CDS/ISIS. Data bases created with the latter operates without change under the Windows version. However, in order to take advantage of the new features (e.g. the graphical or hypertext commands of the formatting language) you may want to review and modify your display formats and/or FST's.

This version of CDS/ISIS for Windows includes all the features of the MS-DOS version and its capabilities are complemented by a number of external programs, such as ImpExp2709 or the XML utilities. It is designed as well as for current MS-DOS users who wish to migrate to the Windows environment, than for new users that will be able to create and manage their own databases.

Finally, the Windows version of CDS/ISIS is better known as “Winisis”. For some reasons some tend to consider the two as totally distinguished programmes. Winisis is one of today's available software based on the CDS/ISIS technology.

Davide Storti
Information Society Division
UNESCO

[1] ISIS_DLL is developed by BIREME/PAHO and UNESCO. Available on UNESCO (www.unesco.org/isis) and BIREME's (www.bireme.br) sites.

[2] In this manual reference is made to the following trademarks:
Microsoft®, MS-DOS® and Windows® are registered trademarks of Microsoft Corporation.

TABLE OF CONTENTS

1. System overview

A. The CDS/ISIS Data Base

CDS/ISIS allows you to build and manage structured non-numerical data bases, i.e. data bases whose major constituent is text.

Although CDS/ISIS deals with text and words, and offers therefore many of the features normally found in word-processing packages, it does more than just text processing. This is because the text that CDS/ISIS processes is *structured* into data elements that you define.

In the most general terms you may think of a CDS/ISIS ***data base*** as a file of related data that you collect to satisfy the information requirements of a given user community. It may be for example a simple file of addresses or a more complex file such as a library catalogue or a directory of research projects. Each unit of information stored in a data base consists of discrete ***data elements***, each containing a particular characteristic of the entity being described. For example, a bibliographic data base will contain information on books, reports, journal articles, etc. Each unit will, in this case, consist of such data elements as author, title, date of publication, etc.

Data elements are stored in ***fields***, each of which is assigned a numeric ***tag*** indicative of its contents. You may think of the tag as the name of the field as it is known by CDS/ISIS.

The collection of fields containing all data elements of a given unit of information is called a ***record***.

The unique characteristic of CDS/ISIS is that it is specifically designed to handle fields (and consequently records) of varying length, thus allowing, on the one hand, an optimal utilization of your disk storage and, on the other, a complete freedom in defining the maximum length of each field.

A field may be optional (i.e. it may be absent in one or more records), it may contain a single data element, or two or more variable length data elements. In the latter case the field is said to contain ***subfields***, each of which is identified by a 2-character ***subfield delimiter*** preceding the corresponding data element. Furthermore a field may be ***repeatable***, i.e. any given record may contain more than one instance or ***occurrence***, of the field.

B. System functions

The major functions provided by CDS/ISIS allow you to:

- Define data bases containing the required data elements
- Enter new records into a given data base
- Modify, correct or delete existing records
- Automatically build and maintain fast access files for each data base in order to maximize retrieval speed
- Retrieve records by their contents, through a sophisticated search language
- Display the records or portions thereof according to your requirements
- Sort the records in any sequence desired
- Print partial or full catalogues and/or indexes
- Develop specialized applications using the CDS/ISIS integrated programming facility.

C. Data base structure

Although a CDS/ISIS data base will appear to you as a single file of information, in actual fact it consists of a number of logically related but physically distinct computer files. The management of the physical files is the responsibility of CDS/ISIS and you do not normally have to know their structure in detail in order to operate a data base. However some basic knowledge of the purpose and function of the major files associated with a data base will help you to understand the system better.

1. Data base definition files

Before a data base can be accessed for processing, it must be made known to CDS/ISIS by defining certain characteristics of its record structure and contents. The Data base definition services allow you to create and/or modify a ***data base definition***.

A CDS/ISIS data base definition consists of the following components, each stored in a separate file:

Field Definition Table (FDT) : The FDT defines the fields which may be present in the records of the data base and their characteristics.

Data entry worksheet(s) (FMT) : One or more screen layouts used to create and/or update the master records of the data base. CDS/ISIS provides a specially designed editor to create these worksheets.

Display format(s) (PFT): Display formats define precise formatting requirements for either on-line display of records during searching or for the generation of printed output products such as catalogues and indexes. CDS/ISIS provides a powerful and comprehensive formatting language which allows you to display the contents of a record in any desired way.

Field Select Table(s) (FST) : One FST defines the fields of the data base to be made searchable through the Inverted file. Additional FSTs define the most frequently used sorting requirements for the data base.

2. Master file

The Master file contains all the records of a given data base, each record consisting of a set of variable length fields. Each record is identified by a unique number, automatically assigned by CDS/ISIS when it is created, called the Master File Number or ***MFN***.

In order to provide a fast access to each master file record, CDS/ISIS associates a special file to the Master file, called the Cross-reference file, which is in fact an index giving the location of each record in the Master file.

You may create, modify or delete Master file records by means of the CDS/ISIS Data Entry services.

3. Inverted file

Although a master record can be retrieved directly by its MFN, through the Cross-reference file, additional ways of accessing a record are, of course, necessary. In the retrieval of bibliographic records, for example, it may be desirable to access a record by author, by subject, or by any other data element occurring in the record. CDS/ISIS allows you to provide a virtually unlimited number of access points for each record through the creation of a special file called the ***Inverted file***.

The Inverted file contains all terms which may be used as access points during retrieval for a given data base, and, for each term, a list of references to the Master file record(s) from which the term was extracted. The collection of all access points for a given data base is called the ***dictionary***. You may think of the Inverted file as an index to the contents of the Master file.

For example, four master records (with MFN 18, 204, 766 and 1039) contain the keyword ADULT EDUCATION. The logical structure of the corresponding Inverted file entry would be:

```
ADULT EDUCATION   18   204   766   1039
```

Here, ADULT EDUCATION is the access point (or dictionary term), and each reference to the Master file record where it appears is called a ***posting***.

Because each term will normally have a different number of records indexed under it, the logical records in an Inverted file are of varying length. Here again, in order to provide the fast retrieval of each access point, the Inverted file actually consists of several physical files.

CDS/ISIS allows selective creation of Inverted files for each data base. You may select fields, subfields or elements thereof. In addition, by specifying appropriate options, you may extract individual words, phrases or descriptors from selected fields.

You define the searchable elements for a given data base by means of a ***Field Select Table*** (FST), which contains the fields to be inverted and the indexing technique to be used for each field.

Unlike other Inverted file based retrieval systems, in which there is a separate Inverted file for each searchable field, CDS/ISIS uses a single Inverted file for any given data base. Because of the particular structure of this file, however, it is functionally equivalent to a multiple Inverted file approach. In actual fact, each posting contains not only the MFN, but also additional information precisely identifying the field from which the data was extracted, as well as the relative word position within the field.

The current implementation provides for access points of up to 30 characters. Elements longer than the maximum length are truncated before an entry is made in the Inverted file.

4. ANY file

An optional type of file, associated with the Inverted file, is the ***Any File***. It is used in retrieval to link together certain related terms. An "***any term"*** is a collective name assigned to a table of search terms. When an ANY term is used in a search, the table with that name is retrieved, and the individual terms in the table are automatically grouped together.

The criterion for the establishment of an ANY term is the likelihood of its frequent occurrence in queries. Geographic groupings will, in most cases, meet this criterion, but other types may qualify equally well. If, for example, you create an ANY term **ANY Latin America**, which defines the names of all the countries in Latin America, you may then use this collective name in a search, rather than typing all the names of the various countries.

5. Relationships between the files

The logical relationship between the major files of a CDS/ISIS data base is best perceived by examining the way in which retrieval is performed. Retrieval from a data base is done by specifying a set of search terms which are looked up in the Inverted File to locate the list of MFNs associated with each term. These lists are then manipulated by the program according to the search operators you have specified in your search formulation until, at the end of the search, a single list, called the ***hit list***, is obtained, corresponding to the MFNs of the records satisfying your search formulation. If at this point you request a display of the records retrieved, CDS/ISIS will read each record in the hit list from the Master file, format it according to the specified format and display it on the screen.

You may also save one or more hit lists, which you may later use to print the records using the Print Dialog services. A saved hit list is called a ***save file***.

D. System architecture

The basic component of CDS/ISIS is its menu system, which allows you to call upon the various services. However, in order to manage and operate your data bases you must also learn a number of ***techniques*** which are specific to CDS/ISIS, such as the search language or the formatting language. Techniques are in turn implemented by using a set of ***tools*** which CDS/ISIS provides for this purpose. For example, if you want to carry out a search in a data base, you must first select the appropriate commands in the menus and then formulate your search requirements, which must follow the rules of the CDS/ISIS search language. You must therefore know this technique. To actually enter the search you use a tool called the "search window".

Whereas a technique entails the intellectual process of transforming a requirement (such as retrieving information on the effects of solar radiation on marine fauna) into the specific search language of CDS/ISIS, a tool is a more mechanical and generally more widely applicable facility (for example the editor is not only used to enter search formulations but also to create or modify records).

1. Menus

You select the operations to be performed by choosing the relevant command from ***menus*** that the system displays on the screen. A command is an instruction that tells CDS/ISIS to perform a certain operation. A menu is a list of commands from which you will make your choice. Menus are displayed in the menu bar at the top of the CDS/ISIS window, just below the title bar. To select a command, first click on the appropriate menu with the left mouse button (this will display the commands available on the selected menu). Then point to the desired command and click the left mouse button.

Note that the various menus may in actual fact be different from the ones displayed in this manual, as each user may change the layout of menus, as well as the corresponding command descriptions. When a menu is displayed, some commands may appear dimmed (i.e. displayed in a light gray color). These commands may not be selected, as they are not operational in that particular context. For example, you may not select the Import command from the above menu if no data base has yet been opened. A command followed by an ellipsis (...) indicates that additional information is required to execute the function. In this case clicking on the command will display a dialog box where this information will be provided.

As an example the Data base menu is reproduced below:

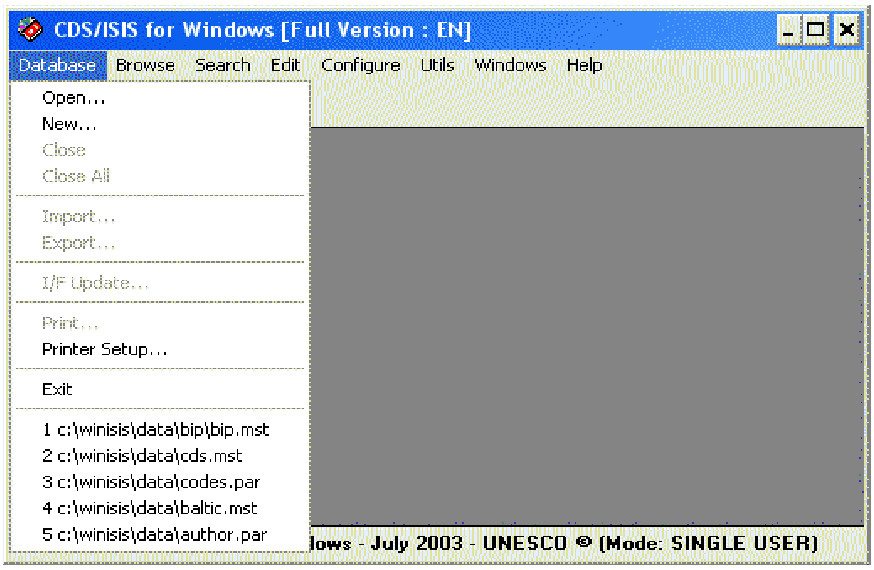

CDS/ISIS for Windows [Full Version : EN]
Database
Browse
Search
Edit
Configure
Utils
Windows
Help
Open...
New...
Close
Close All
Import...
Export...
I/F Update...
Print...
Printer Setup...
Exit
1 c:\winisis\data\bip\bip.mst
2 c:\winisis\data\cds.mst
3 c:\winisis\data\codes.par
4 c:\winisis\data\baltic.mst
5 c:\winisis\data\author.par
lows - July 2003 - UNESCO © (Mode: SINGLE USER)

Figure 1 - Sample CDS/ISIS menu

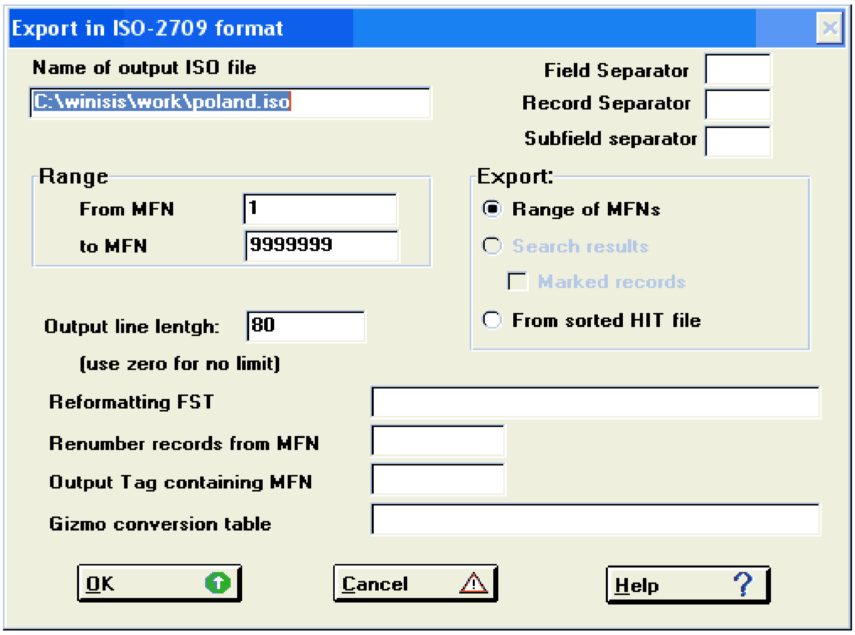
2. Multilingual dialogue

CDS/ISIS is fully interactive and multilingual. The latter facility is particularly interesting as it allows you to select the language in which the system menus and messages will be displayed.

Each time it starts CDS/ISIS displays the menus in the default language selected by your data base manager when the system was installed on your machine. You may change the dialog language at any time, by executing the ***Change Language*** command in the **Configure** menu. This will display the list of available languages. To select a language, click on the desired one with the left mouse button. From that point onwards all menus, system messages and prompts will be in the language you have chosen. Note, however, that any open windows will remain in the language being used when they were created.

3. Dialog boxes

In some cases CDS/ISIS needs additional information before it can execute a command. You provide this information by selecting options in a special type of window called a *dialog box*. For example, the Export dialog box is shown below:



Export in ISO-2709 format

Name of output ISO file: C:\winisis\work\poland.iso

Field Separator

Record Separator

Subfield separator

Range

From MFN 1

to MFN 9999999

Export:

● Range of MFNs

○ Search results

□ Marked records

○ From sorted HIT file

Output line lentgh: 80

(use zero for no limit)

Reformatting FST

Renumber records from MFN

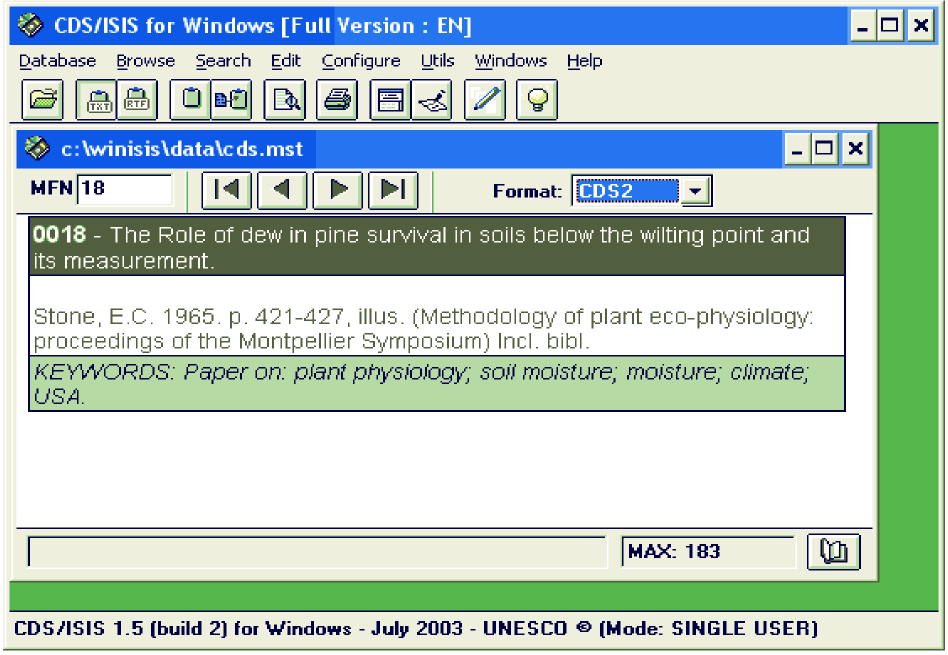
Output Tag containing MFN

Gizmo conversion table

Figure 2 - Sample CDS/ISIS Dialog Box

4. Windows

CDS/ISIS uses different types of **windows** which are designed for a specific purpose. For example, to display a record of a data base it will use the **data base window** (see the sample below), to carry out a search the **search window**, or to enter data in a record the **data entry window**. These are automatically put on the screen as soon as you select the corresponding menu command.



CDS/ISIS for Windows [Full Version : EN]

c:\winisis\data\cds.mst

MFN 18

Format: CDS2

0018 - The Role of dew in pine survival in soils below the wilting point and its measurement.

Stone, E.C. 1965. p. 421-427, illus. (Methodology of plant eco-physiology: proceedings of the Montpellier Symposium) Incl. bibl.

KEYWORDS: Paper on: plant physiology; soil moisture; moisture; climate; USA.

MAX: 183

CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: SINGLE USER)

Figure 3 - Sample Data base window

2. System installation

A. Hardware requirements

The minimum and recommended hardware requirements for running CDS/ISIS are the following:

CPU:	486 processor at 40Mhz (Pentium at 100Mhz or higher recommended)
RAM:	8Mb (16Mb or more recommended)
1	Floppy or CD-Rom unit
1	hard disk (with at least 4Mb free)
1	VGA 640x480 color screen (super VGA 800x600 or higher recommended)
1	printer (optional)
Windows 3.1 or higher	

Note: *Although CDS/ISIS is a Windows 3.1 based program, it runs under Windows95, 98, NT4, 2000 and XP with no specific problems. (see p.13-14). Under Windows Vista, ensure that all the correct file and folder access permissions are set, and that 16bit programs are supported.*

B. Installing CDS/ISIS on your computer

1. CDS/ISIS setup

CDS/ISIS may be distributed in different ways:

- Through the Internet: for example at http://www.unesco.org/isis
- On the UNESCO's Information Processing Tools CD-Rom.
- on 2 diskettes labelled **ISIS01** and **ISIS02**. The first contains the software and the second contains the documentation.

Whatever is the format you have got CDS/ISIS, there will be a file called either "**WINISIS.EXE", "WINISIS15.EXE"**, "**SETUP.EXE" or a ZIP file** containing the CDS/ISIS installer. Run that file by double-clicking it or through the Start-Run Windows menu.

The setup program will ask you a number of questions on the placement of the various system files.
If this is your first installation of the system, **we recommend** you accept the proposed defaults.

Most of the parameters will be used to create the system parameter file **SYSPAR.PAR** described under "1. SYSPAR.PAR: Global parameters". The information you will be asked to provide is as follows:

- The main CDS/ISIS directory (by default **\winisis**).
- The Windows Program Manager Group where the CDS/ISIS icons will be placed (by default **Winisis**).
- The data base directory, i.e. the directory where your data bases are or will be stored. For compatibility with the MS-DOS version of CDS/ISIS the default is **\winisis\data**. The test CDS data base supplied will be installed in this directory.
- The initial dialogue language (by default **EN**). CDS/ISIS, as provided by UNESCO, supports the following languages: **EN** (English), **FR** (French), **SP** (Spanish). However, copies of CDS/ISIS provided by National distributors may support additional languages.

- The CDS/ISIS Pascal program directory, i.e. the directory where your CDS/ISIS Pascal programs are or will be stored. For compatibility with the MS-DOS version of CDS/ISIS the default is **\winisis\prog**. The sample programs supplied will be installed in this directory.

2. CDS/ISIS directories

The installation procedure described above will create a main CDS/ISIS directory and a number of subdirectories, each containing a specific category of files, as follows (default names are given in parentheses).

Main directory (\winisis): contains the executable file (**WISIS.EXE**) and related files. You must not remove or modify any of the files installed in this directory other than the **SYSPAR.PAR** file described below.
Program subdirectory (\winisis\prog): contains application programs written in CDS/ISIS Pascal.
Menu subdirectory (\winisis\menu): contains menus data bases and system tables.
Message subdirectory (\winisis\msg): contains system messages data bases.
Work files subdirectory (\winisis\work): contains workfiles generated by CDS/ISIS.
Data base subdirectory(\winisis\data): contains data base files (this subdirectory may itself contain other subdirectories as explained under "**Data base parameters**" on page 26.
Image files subdirectory (\winisis\bg): contains some Winisis background images and serve as the default directory for images if these are provided without a pathname.

3. ISISPAS.PIF

The setup program installs the file ISISPAS.PIF in your Windows directory and the file ISISPAS.EXE (the CDS/ISIS Pascal compiler) in your main CDS/ISIS directory. The PIF file is required to compile CDS/ISIS Pascal programs with CDS/ISIS. It contains the path to the ISISPAS.EXE file and the name of the work directory to be used. If you installed CDS/ISIS in a directory other than **\winisis** you must edit this file to change the defaults accordingly.
Under Windows 3.x proceed as follows:

- From Program Manager call the PIF Editor (in the Main window);
- Open the ISISPAS.PIF file;
- Modify as appropriate the directory in the **Program Filename** and in the **Start-up Directory** boxes;
- Close and save the file.
- Under Windows 95 proceed as follows:
- Locate the file ISISPAS.PIF using Windows Explorer;
- Point to the file, click the right mouse button and select Properties;
- Select the Program tab;
- Modify the directory in the **Cmd line** and in the **Working** boxes as appropriate;
- Close and save the file.

However, the PIF file may not work under Windows 2000 or XP.

4. Windows versions compatibility

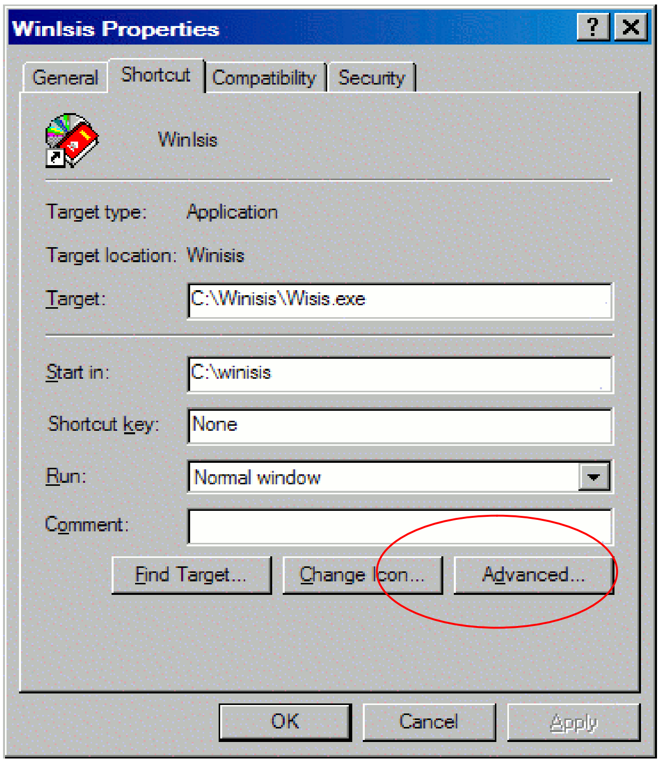
As mentioned earlier, CDS/ISIS is known to run under all Windows versions without problems. In some cases, however, it may be necessary to install an extra file, if you experience problems such as buttons or windows incorrectly displayed. To solve these problems you should copy the files CTL3D.DLL and CTL3DV2.DLL (which have been stored in the subdirectory **ctl3d** of the main CDS/ISIS directory by the setup procedure) to your **\windows\system** directory. (or \winnt\system for WinNT and Win2000 users).

WARNING: Long file names are NOT supported for data base files. Therefore only 8-character directory names and, for compatibility with the MS-DOS version, 6-character data base names are accepted (without spaces).

This is particularly important for data base and system directories. It is again **recommended** to use default path names as suggested during the installation procedure.

Last but not least, under Win2000 and WinXP it is recommended to launch Winisis from a shortcut, instead of its executable file. This is because the shortcut allows you to fine tune the session in which Winisis will be run:

- Create a **shortcut** to file WISIS.EXE on your desktop
- Right-click on the shortcut and select "**properties**".



WinIsis

Target type: Application

Target location: Winisis

Target: C:\Winisis\Wisis.exe

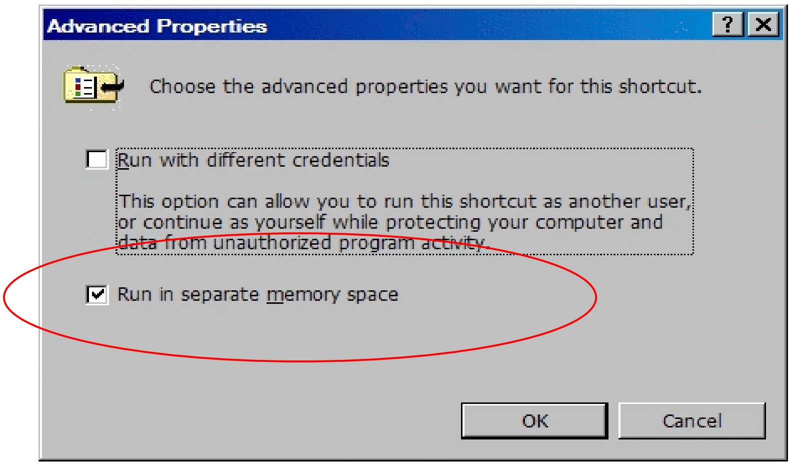
Start in: C:\winisis

Shortcut key: None

Run: Normal window

Comment:

- Look for and click the option "Run in separate memory space". On WinXP click the button for "Advanced options" to find such an option.



Choose the advanced properties you want for this shortcut.

□ Run with different credentials

This option can allow you to run this shortcut as another user, or continue as yourself while protecting your computer and data from unauthorized program activity.

☑ Run in separate memory space

C. System Parameter files

1. SYSPAR.PAR: Global parameters

The system file **SYSPAR.PAR** contains system setup parameters. It is read each time CDS/ISIS is activated and may be used to override default values which the system would otherwise apply. When you initially installed CDS/ISIS, the setup program automatically created a **SYSPAR.PAR** file with the default parameters you supplied. This section describes the format and contents of this file (note that parameters 1-8 are the same as in the MS-DOS version).

SYSPAR.PAR is a text file which may be edited with any text editor such as **Notepad**. Each parameter starts on a new line and has the following general format:

n=value

where:

n is the parameter number (as explained below);

value is the corresponding value

The value must immediately follow the equal sign (any space which may follow the equal sign will be taken as being part of the value). When the value consists of a directory name, this will be concatenated with the CDS/ISIS file name to build the actual file specification. Therefore the directory name must be terminated by a backslash or else an invalid file name will be generated. For example:

```
2=\winisis\menu\
```

Missing parameters, or parameters with a blank value will be assigned the standard system default. Therefore **SYSPAR.PAR** need only contain those parameters which you wish to change. The parameters may be given in any order.

You may insert *comment* lines (which will be ignored by CDS/ISIS) by inserting a semicolon (;) at the beginning of the line. For example:

```
; Set default database
```

Note that comment lines might be destroyed if you edit the system parameter from within Winisis.

The complete list of parameters which you may specify is given below. Note, however, that only parameters 1 to 5 are normally required, while other parameters may be used to alter the default behavior of CDS/ISIS, but are not essential.

Parameter 0: SYSPAR.PAR re-direction

This parameter may be used to redirect the **SYSPAR.PAR** file itself to another drive or directory. The value may be:

1. a full file name, which may optionally include drive and/or directory information; for example:

   ```
   0=c:\isis\mypar
   ```

   in this case the actual parameters to be used will be read from the file **mypar.par** in the directory \isis of drive c:;

2. a question mark (?) or an exclamation mark (!) followed by a prompt; for example:

   ```
   0=?Enter user name:
   ```

in this case CDS/ISIS will first display the prompt and then read the name of the parameter file from the keyboard. You may use this form of the parameter to define the name of the parameters file each time you call the program from a given directory. If you use an exclamation mark (!) instead of a question mark, the text typed in response to the prompt will not be echoed to the screen.

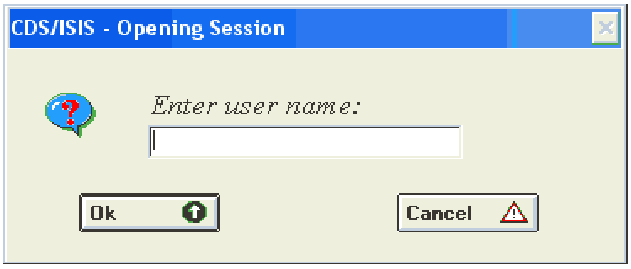


Figure 4 - Syspar parameter 0 example

SYSPAR.PAR may be re-directed several times if required. For example, the file MYPAR.PAR in the example above may itself contain a parameter 0 specifying another file. Note that if you re-direct SYSPAR.PAR, and the new file defines any parameter already defined in a previously processed parameter file they will be overridden by the new value. Furthermore, any parameters following parameter 0 in the current file are ignored.

You may use parameter 0 to provide a certain amount of system and data protection, by defining a SYSPAR.PAR containing generally applicable parameters and one file for each user or group of users which may be empty or possibly containing selected parameters specific to each user. Suppose for example you have created the following files:

```
  SYSPAR.PAR            JOHN                   DAVID
--------------    ------------------    -------------------
1=\isis\prog\     4=\isis\john\work\    4=\isis\david\work\
2=\isis\menu\     5=\isis\john\data\    5=\isis\david\data\
3=\isis\msg\      6=BIBLIO
101=SP            101=EN
0=!password...
```

In this case, each time CDS/ISIS is started it will first read **SYSPAR.PAR** and set the system, menu and message paths and select Spanish as the default language. Then the user will be prompted to enter the password (parameter 0). If the response to this prompt is **john**, CDS/ISIS reads this file and sets the remaining parameters: this user, therefore, will work in English and use BIBLIO as his default data base.

For better security you may *hide* these files by setting the appropriate file attribute.

Parameter 1: Program path

This parameter defines the drive and/or directory where CDS/ISIS Pascal application programs are stored (**PAS** and **PCD** files). By default, these are assumed to be in the current directory. Note that program listing files (**LST**) are stored in the work files path defined in parameter 4.

Parameter 2: Menus path

This parameter defines the drive and/or directory where the CDS/ISIS menu files are stored. By default, menus are assumed to be in the current directory. All files associated with a menu data base (**xxMNyy.***) will be assigned to this path.

The files **ISISUC.TAB** and **ISISAC.TAB** must also be in this directory.

Parameter 3: Message path

This parameter defines the drive and/or directory where the CDS/ISIS system message files are stored. By default they are assumed to be in the current directory. All files associated with a message data base (**xxMSG.***) will be assigned to this path.

Parameter 4: Work files path

This parameter defines the drive and/or directory where CDS/ISIS will create any required work file (e.g. sort work files). By default, work files are created in the current directory. All files not covered by other parameters will be assigned to this path, e.g. print files.

Parameter 5: Data base path

This parameter defines the drive and/or directory where data base files are stored. By default, data base files are assumed to be in the current directory.

The following data base files (except for message data bases) will be assigned to this path: **ANY, CNT, IFP, L01, L02, N01, N02, MST, XRF, FDT, FST, FMT** (data entry worksheets), **PFT, STW** and **SRT**. All other data base files such as **HIT, LN?, LK?** etc. are assigned to the work files path specified in parameter 4.

Note that Data base files may themselves be distributed over one or more directories and/or drives by supplying a **dbn.PAR** file (see under "Data base parameters" below).

Parameter 6: Default data base name

This parameter defines the name of the default data base. If specified, the default data base will be automatically selected upon program initiation. It is particularly convenient for users normally working on a given data base.

Parameter 8: Repeatable fields separator

This parameter defines the character to be used by CDS/ISIS to separate the occurrences of a repeatable field during data entry. By default CDS/ISIS will use a percent sign (%), which effectively reserves its use for this purpose. If you need to enter percent signs as data, you may define here another character to be used instead. Note that, as this character becomes in turn a CDS/ISIS control character, you should choose one which you are sure will never occur in your data. For example:

```
8=&
```

defines the character **&** as the repeatable fields separator.

Note that while the MS-DOS version of CDS/ISIS allows only a single character for parameter 8, the Windows version allows you to specify a string of characters. For example:

```
8=$$$
```

Parameter 14: Network mode

Winisis may work on Windows networks, provided that these are already configured in the hosting machine. Supported modes are:

14=0 → single user mode
14=1 → multi user mode

For more information, please read the corresponding documentation available on the UNESCO CDS/ISIS web page.

Parameter 101: Default language

This parameter is a two-letter code defining the initial dialogue language to be used. The initial default language is **EN** (English).

Parameter 102: Default menu

This parameter is a two-letter code specifying the default menu to be used. It allows you to setup limited profiles of CDS/ISIS. The parameter gives the last two letters of the menu data base to be used. By default these are **DF**. For example, if neither parameter 101 or 102 are present, the default menu data base will be **MNENDF**, whereas if you specify **102=SH** it will be **MNENSH**. In this case, of course, you must have previously created this data base.

Parameter 103: Display search details

The value of this parameter can be **0** (default) or **1**. When set to **1** CDS/ISIS will display a results window showing the progress of a search. When set to **0** this window is not displayed.

Parameter 104: Display dictionary options

The value of this parameter can be **0** (default) or **1**. It affects the display of the dictionary window when a field is selected. When set to **0** (default) CDS/ISIS displays the complete dictionary, however, terms not occurring in the selected field are displayed in lower case (and will not be selectable). When set to **1** only terms occurring in the selected field are displayed.

Parameter 105: Automatic display of search results

The value of this parameter can be **0** (default) or **1**. When you are using expert search mode and this parameter is set to **1** CDS/ISIS automatically displays the results of a search as soon as it is executed. When set to **0** (default) CDS/ISIS remains in search mode.

Parameter 106: DOS to Windows conversion table

This parameter specifies the file name of a conversion table that CDS/ISIS will use to convert characters from DOS to Windows. By default the standard Windows conversion table is used. See under "B. Conversion tables" on page 140 for more details on this option.

Parameter 107: Windows to DOS conversion table

This parameter specifies the file name of a conversion table that CDS/ISIS will use to convert characters from Windows to DOS. By default the standard Windows conversion table is used.

Parameter 108: Screen orientation

The value of this parameter can be **0** (default) or **1** and defines the default screen orientation. When set to **0** the orientation will be left to right, while when set to **1** (e.g. for the Arabic version), the orientation will be right to left.

Parameter 109: Field edit font

This parameter specifies the font and/or font size to be used in the field edit box of the data entry window. The general format of this parameter is:

109=*font Name*[,*font size*]

where:

font name	is the name of the font. It must be a font known to Windows (you may check this with a Word processor or by using the Fonts option of the Windows Control Panel)
font size	is the font size in half points

For example:

```
109=Arial Narrow       specifies the "Arial Narrow" font with the
                       default font size
```

```
109=Arial,24           specifies the "Arial" font with a 12 point font
                       size.
```

Note that CDS/ISIS sets this parameter automatically whenever you change the font in the Data Entry Window

Parameter 110: Formatting defaults

This parameter may be used to override the default font and color tables used to display data base records. The general format of this parameter is:

110=format

where **format** is a CDS/ISIS format which may only contain the fonts, cols, cf, f, fs in this sequence. The default for this parameter is:

```
110=fonts((nil,Courier New), (swiss,Arial)), cols((0,0,0)), CL0, F0, FS24
```

Note that this parameter must be contained in a single line. Detailed explanations on the above formatting commands are given under "N.	The Windows graphic environment".

Parameter 111: Format edit font

This parameter specifies the font and/or font size to be used in the format edit box of the data base window. The general format of this parameter is the same as Parameter 109.

Parameter 112: Printer set up

This parameter is automatically set by CDS/ISIS. It stores the name and port of the latest printer you used. You must not create or edit this parameter directly, but you may modify it by using the ***Printer setup*** command of the **Data base menu**.

Parameter 115: Small font definition

This parameter defines which font should be used as "small" font for some of Winisis' windows, such as the Dictionary and the Search windows. This parameter shall be set using the System settings dialog box. The format is:

115=Font name, size

See parameter 109 for examples.

Parameter 116: Dialogs font definition

This parameter defines which font should be used as the default font for Winisis' dialogs, such as the "Open data base" and Import/Export windows. This parameter shall be set using the System settings dialog box. The format is:

116=Font name, size

See parameter 109 for examples.

Parameter 117: System font definition

This font is used for displaying system text such as "group descriptions" and it is usually bigger than the dialog font. This parameter shall be set using the System settings dialog box. The format is:

117=Font name, size

See parameter 109 for examples.

Parameter 120: Display Toolbar

The value of this parameter can be **0** or **1** (default). When set to **1** CDS/ISIS will display a Toolbar under the menu bar (a Toolbar consists in a number of buttons giving fast access to the most frequently used functions of CDS/ISIS). When set to **0** the Toolbar is not displayed.

Parameter 121: Display Toolbar Help

The value of this parameter can be **0** or **1** (default). When set to **1** CDS/ISIS will display a Help message (at the bottom of the screen) each time the mouse passes over a button of the Toolbar indicating its function. When set to **0** the Help message is not displayed.

Parameter 122: CDS/ISIS Window position

This parameter is used by CDS/ISIS to store the latest status of the main window. You should not create or edit this parameter.

Parameter 123: Show empty fields

The value of this parameter can be **0** (default) or **1**. In data entry mode, when this parameter is set to **1** CDS/ISIS will show all the fields indicated in the data entry worksheet, including empty fields. When this parameter is set to **0** only fields containing data will be displayed.

Parameter 124: Database window frame

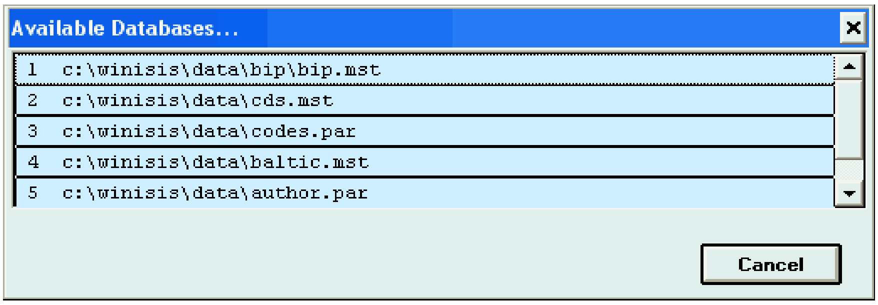
This parameter may alter the default dimension and the position of the data base window. The general format is:

124=NoToolbar,left,top,right,bottom

NoToolbar (default is 0) will hide data base window's toolbar is set to 1.
Left, **Right**, **Top** and **Bottom** define the position and the dimensions of the data base window.
If **Right** or **Bottom** values are set to **-1**, the window will appear maximized.

Parameter 125: Data base selection mode

This parameter affects the way a user will be able to choose the database to open. Default is 0. When 125=1, the user will only be able to select databases which are listed under parameters 1001 to 1005.



```
1   c:\winisis\data\bip\bip.mst
2   c:\winisis\data\cds.mst
3   c:\winisis\data\codes.par
4   c:\winisis\data\baltic.mst
5   c:\winisis\data\author.par
```

Figure 5 - Syspar parameter 125=1 example

Parameter 130: Automatic Inverted file update

The value of this parameter can be **0** (default) or **1**. When closing the data entry window and this parameter is set to **1** CDS/ISIS will automatically update the Inverted file. When set to **0** the updating of the Inverted file must be manually requested by means of the ***I/F Update*** command of the **Data base** menu.

Parameter 131: Word length

This parameter defines the maximum length of a word. This allows Winisis to handle non Latin scripts, such as Chinese, that do not use spaces to separate words. For example, setting **131=10**, will cause a line break at the 10th character, whenever necessary, to display or paginate the text.

Parameter 132: CDS/ISIS Pascal error

The value of this parameter can be **0** (default) or **1**. When set to **1** CDS/ISIS displays a dialog box whenever the CDS/ISIS Pascal interpreter detects an error. When set to **0** the dialog box is not displayed. In both cases, however, the error message is displayed in the output text produced by the format.

Parameter 133: Windows system font

Windows allows you to use two types of system font: small or large. By default, CDS/ISIS is configured to run with the small font configuration. However, if your Windows is configured to use large fonts, many windows will be unreadable, with misplaced buttons or with no buttons at all. If this is the case, you must set parameter 133 to tell CDS/ISIS the system font currently in use

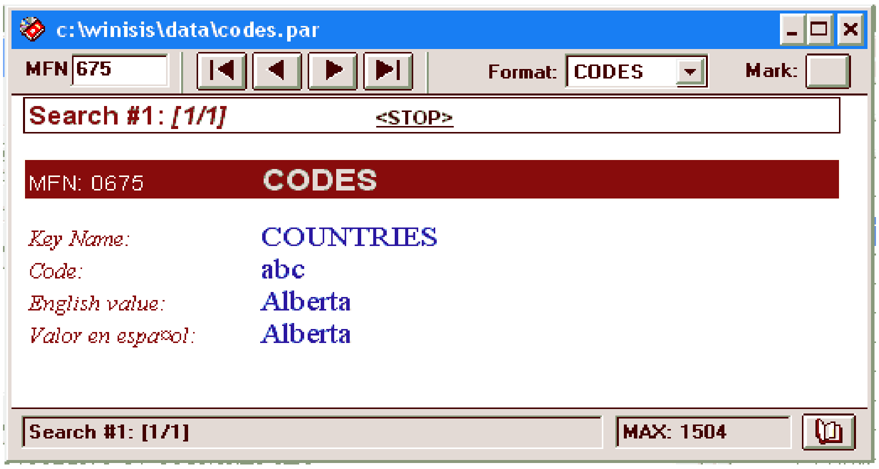
The value of this parameter can be **0** (default) or **1**. When set to **0** CDS/ISIS assumes you are using small fonts, when set to **1** it assumes you are using large fonts. You must change the value of this parameter each time you change your Windows settings.

Parameter 134: Data entry labels width

This value sets the width of the column of field names/description in the data entry window. Format: 134=width in pixels

Parameter 135: Alternate search results toolbar

Setting this parameter to 1 will display a hypertext on the top of each record while browsing a search result. A <STOP> hyperlink lets the user return browsing the full data base.



c:\winisis\data\codes.par

MFN 675

Format: CODES

Mark:

Search #1: [1/1] <STOP>

MFN: 0675 CODES

Key Name: COUNTRIES

Code: abc

English value: Alberta

Valor en espa¤ol: Alberta

Search #1: [1/1]

MAX: 1504

Figure 6 - Alternate search results toolbar

Parameter 136: Record display zoom

It is possible to zoom (in or out) the content of a data base window using this parameter. Format: **136=zoom percentage** (80-160)

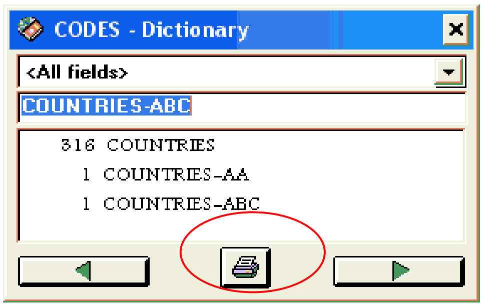
136=140

Parameter 137: Close Data Base confirmation

Setting this parameter to 1 will ask for confirmation before closing a data base window.

Parameter 138: Enable/disable Inverted File dump

When set to 1, the dictionary window will display a button allowing the dump of the Inverted File.



```
316 COUNTRIES
  1 COUNTRIES-AA
  1 COUNTRIES-ABC
```

Figure 7 - Dictionary window with Dump button

Parameter 140: Display deleted records

This parameter hides, when set to 0, the content of deleted records.

Parameter 141: Pattern/Images path

This parameter defines the default folder for patterns/images. Default is c:\winisis\bg\
It is used if no path is defined for background images and patterns.

Parameter 142: Retrieved terms highlighting (and colour)

This parameter sets the foreground and background colours of retrieved terms after a search.

Parameter 144: Force results display in "listing mode"

Winisis may display search results in a list. If only one term is retrieved, however, it is usually displayed in a single record window. Setting 144=1 will force the display in a list (thus containing only one item).

Parameter 150: Date format

This parameter sets how command DATE will display the current system date. Available formats:
150=YMD (Year, Month, Day)
150=DMY (Day, Month, Year)
150=MDY (Month, Day, Year)

Parameter 152: Close print dialog after job is completed

Setting this parameter to 1 will automatically close the print dialog once the printing is finished. Default is 0.

Parameter 155: Data Entry. Repeatable subfields

This parameter enables data entry checking for repeatable subfields:
155=0 → don't check
155=1 → check and show errors

Parameter 156: Data Entry. Record validation

This parameter determines how Winisis will perform record validation:
156=0 → Validate only modified records
156=1 → Always validate records
Validation rules are stored in data base's VAL file.

Parameter 157: Data Entry. Enable field type and pattern control

Field definition (FDT) contains information about each field’s data type and its data entry mask (pattern). For example:
Field Code (tag=20, type=alphanumeric, pattern=XX99-AA)
This parameter enables/disables control on recognized field types (alphanumeric, alphabetic, numeric) and patterns.
Patterns apply to each type character. Wildcards are:

X	accepts alphanumeric characters (all chars)
A	accepts alphabetic characters only
9	accepts numeric characters only
Any other char.	Mandatory in the given position

Parameter 160: Start-up plug-in(s)

Define the external program(s) to be run when Winisis starts. This parameter is repeatable. Example:

```
160=mystart.exe
160=example.exe
```

The programs should be located in Winisis’ plug-in folder. Winisis passes to these programs three parameters:

frameID	Winisis’ window handle
workpath	Winisis’ default work path
version	Winisis’ version number (e.g. 15003 or 14019)

Parameter 161: Shutdown plug-in(s)

Define the external program(s) to be run when Winisis is closing. This parameter is repeatable. Example:

```
161=closeall.exe
161=dosomth.exe
```

The programs should be located in Winisis’ plug-in folder. Winisis passes to these programs three parameters:

frameID	Winisis’ window handle
workpath	Winisis’ default work path
version	Winisis’ version number (e.g. 15003 or 14019)

Parameter 902: Debug hypertext commands

Setting this parameter will allow print format hypertext debugging: before executing any hypertext command, the command itself will be shown on the screen.

Parameter 910: HELP files path

Winisis 1.5 uses HTML help files (if present). This parameter sets the default folder where these files are located.

Parameter 911: HELP files prefix

Winisis 1.5 HTML help files names are determined as follows:
[path][prefix][code][language_code]
Codes are fixed and language codes comes from the active menu.

Window	Winisis 1.4 code	Winisis 1.5	
		Code	Filename example (EN = English)
Data entry	3	DE	`ISISDEEN.HTM`
Direct (file open)	9	FL	`ISISFLEN.HTM`
Easysearch	5	ES	`ISISESEN.HTM`
Logic1	4	SR	`ISISSREN.HTM`
Print #1 General	10	PG	`ISISPGEN.HTM`
Print #2 Presentation	11	PP	`ISISPPEN.HTM`
Print #3 Margins	12	PM	`ISISPMEN.HTM`
Print #4 Layout	13	PL	`ISISPLEN.HTM`
Wisis #1 HelpIndex	0	IX	`ISISIXEN.HTM`
Import	6	IM	`ISISIMEN.HTM`
Export	7	EX	`ISISEXEN.HTM`
Inverted file update	8	IU	`ISISIUEN.HTM`

At this time, however, HTML HELP files are not available in the installation file. Please check UNESCO CDS/ISIS web site for HTML files availability.

Parameter 918: Set ISO default record and field separators

This parameter sets the default record and field separators for the ISO export. Format:
918=AB
“A” will be used as record separator
“B” will be used as field separator

Parameter 939: Winisis Log file

This parameter enables Winisis operations file logging. The format is:
939=filename

The first time Winisis is run on the computer, a **unique identifier** for that machine is created: that ID is used in the log file. The ID is maintained across different Winisis installations on the same machine. The ID is a number. On a Network, it is recommended to use one common log file. It is up to the administrator to keep track of these numbers.

If no path is specified, the database default directory is retained as the directory for the log file.

Currently logged operations are:

```
    - record [Updated  ]
    - record [Created  ]
    - record [Ldelete  ]
    - record [LUndelete]
    - [IF update]
    - [Global delete tag xx]
```

Parameter 945: Create current system's ASCII table

This parameter builds current system's ASCII conversion table. For example, in order to switch from OEM to ASCII and vice-versa, it is necessary to use conversion tables. While this operation is transparent, in some cases it is necessary to specify the conversion to be applied (during the import, for example). Adding this parameter will force Winisis to create 3 tables:

`txt_.tab` – from ANSI to OEM (your Windows' DOS codepage)
`_txt.tab` – from OEM (your Windows' DOS codepage to ANSI)
`_upcase.tab` – the table normally used for conversion to uppercase

The 3 tables are created ***as soon as Winisis starts***. You should **delete** this parameter afterwards. Format:

945=[any character]

Parameter 980: Display search results in a table

This parameter affects the way search results are shown to the user. Setting 980=1 will show the results in a table instead of record by record.

Parameters 1001-1005: Recently used data bases

These parameters are used by CDS/ISIS to store the last 5 databases you opened during previous sessions. These names will appear as commands in the **Data base** Menu. You should not normally set or change these parameters manually.

2. dbn.PAR: Data base parameters

When you open a data base, CDS/ISIS will first try to locate a file called **dbn.PAR** (where **dbn** is the data base name) in the data base path specified in Parameter 5 of SYSPAR.PAR. If this file does not exist all data base files are assumed to be in the data base path.

dbn.PAR allows you to define individual paths for specific data base files and its format is compatible with the MS-DOS version. You may therefore distribute a large data base on two or more disk drives, if necessary. dbn.PAR may contain up to 10 parameters (which have the same format as the ones specified in SYSPAR.PAR) as follows:

```
Parameter       Drive/path for
---------       --------------
    1                XRF
    2                MST
    3                CNT
    4                N01
    5                N02
    6                L01
    7                L02
    8                IFP
    9                ANY
   10                FDT, FST, FMT, PFT, STW, SRT, WPR
```

Not all parameters need be specified, but to avoid confusion it is best to specify all 10. Files for which you do not define a path will be located in the data base path defined in parameter 5 of SYSPAR.PAR. For example, the following parameter file tells CDS/ISIS that the Master file is on drive **d:** and the **IFP** file on drive **e:** on the indicated directories (the other files of the data base will be in the data base path specified in **SYSPAR.PAR**):

```
2=d:\biblio\
3=e:\biblio\
```

The data files for each database should always be kept in its own folder. A common folder tree is something like below (for two databases CDS and BOOKS). The CDS.PAR and BOOKS.PAR files are then in the DATA folder, and parameter 5 in SYSPAR.PAR points to the DATA folder.

```
DATA-|
     |-CDS
     |-BOOKS
```

D. System restrictions

The following system restrictions are currently in effect:

Maximum number of data bases	unlimited
Maximum number of records in a data base	16 millions (within the limit of 500 Mb)
Maximum record size	32000 characters[3]
Maximum field size	32000 characters
Maximum number of fields (defined in FDT)	200 (excluding repetitions of repeatable fields)
Maximum number of FST lines	600
Maximum number of stopwords	799
Maximum size of a display format	26 Kbytes
Maximum size of display buffer	64 Kbytes

[3] Note that records larger than 8000 may not be used with the MS-DOS version of CDS/ISIS. If you intend to exchange data with MS-DOS users you should take this into account.

3. CDS/ISIS Menus

A. The Main Window

This window is displayed each time CDS/ISIS is started. It is normally empty, as shown below, unless a default data base is selected through parameter 6 of the **syspar.par** file. In this case the data base window of the default data base is also automatically opened.

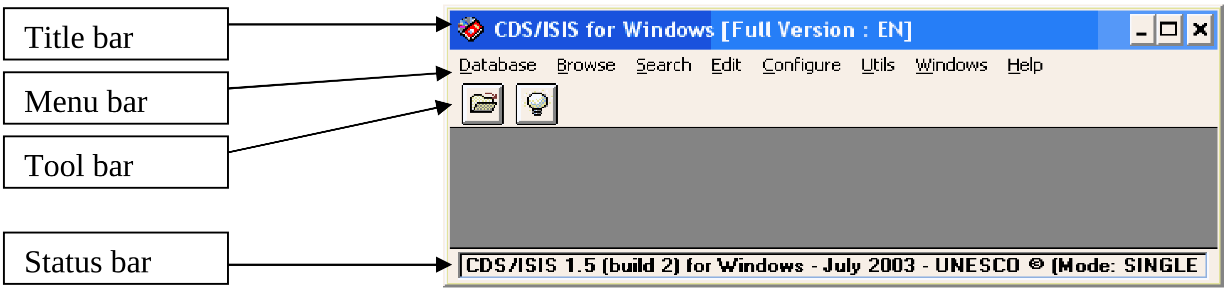

Title bar
Menu bar
Tool bar
Status bar
CDS/ISIS for Windows [Full Version : EN]
Database Browse Search Edit Configure Utils Windows Help
CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: SINGLE

Figure 8 - Main window

The main components of this window are:
the **window title bar**, at the top of the window;
the **menu bar**, which provides access to all the CDS/ISIS functions (some of these functions may also be activated by clicking on the various buttons of the tool bar);
the **tool bar**, located just under the menu, which provides a quick mouse access to the most frequently used functions of CDS/ISIS, such as opening a data base, searching or printing (the tool bar buttons are explained under *The data base window*);
the **status bar**, at the bottom of the window, which is normally used to display help messages. The status bar may be disabled by setting the appropriate option in **syspar.par**.
The following sections describe each menu in detail.

B. Database menu

This menu contains the following commands:

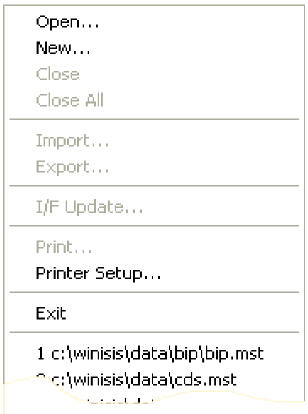


Figure 9 - Database Menu

The bottom part of the menu normally contains the list of the most recently opened data bases. You may open any one of them by simply clicking on the corresponding entry.

i. Open

There are three ways you may open a data base:
the first is by selecting one of the five most recently opened data bases from the list shown at the bottom of this menu;
the second is by selecting this command;
the third is by clicking on the open data base button on the toolbar: 📂
In the last two cases CDS/ISIS will display the **open dialog box**. After opening a data base the corresponding **data base window** is displayed.

ii. New…

Using this command it is possible to create new CDS/ISIS databases, using the Database Definition Wizard, which consists of 4 main steps:
definition of fields
definition of a data entry worksheet
definition of a print format
definition of indexing rules (for searching).

iii. Close

This command closes the currently selected data base. All associated windows, such as a search window, will also be automatically closed.

iv. Close all

This command closes all the currently open data bases.

v. Import

This command allows you to import data from external files recorded according to the ISO-2709 standard format for information interchange. When you select this command CDS/ISIS will first display an **Open Dialog Box**, where you select the file to be imported, and then the **Import Dialog Box** which contains the various parameters you must also supply.

vi. Export

This command allows you to extract a data base or a portion thereof normally for transmitting it to other users. You may also use this command to perform some reformatting of the records of a data base and then use the import function to store the reformatted data into the original or a different data base. When you select this command CDS/ISIS will first display an **Open Dialog Box**, where you select the output file, and then the **Export Dialog Box** which contains the various parameters you must also supply.

vii. I/F Update

This command allows you to perform various maintenance operations on the Inverted file of the selected data base. When you select this command CDS/ISIS displays the **Inverted File Maintenance Dialog Box** which contains the various parameters you must also supply.

viii. Print

This command allows you to print the output of a given query and/or to print a selected range of records. You may sort the records by virtually any combination of fields and subfields. The field(s) by which the records are sorted may be used as headings in printing. When you select this command CDS/ISIS displays the **Print Dialog Box** where you will be asked to provide the specific sorting and page layout parameters you require for that particular print run.

ix. Printer Setup

This command allows you to setup (or verify) the printer options before a print run. When you select this command CDS/ISIS displays the **Print Setup Dialog Box**.

x. Exit

This command terminates CDS/ISIS. All open data bases will automatically be closed.

C. Browse menu

This menu contains the commands shown below, which affect the contents of the **data base window** and the **data entry window**.

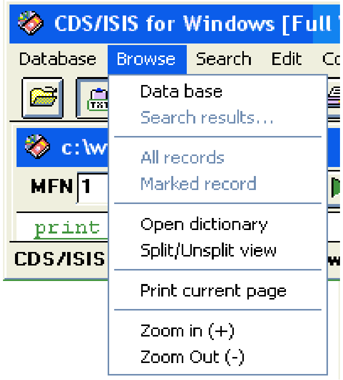


Figure 10 - Browse Menu

i. Data base

Sets the data base in *browse mode*. In this mode the whole data base may be browsed or edited independently from any search performed.

ii. Search results

Sets the data base in *display search mode*. In this mode only the records retrieved by the selected search are displayed. When you choose this command, CDS/ISIS displays the list of searches performed so far, from which you may select the one desired.

You may also use this command to simply view the list of searches submitted so far.

iii. All records

This command is only active when you are displaying or editing search results. It allows you to view all the records retrieved, independently from their marked status. The mark box will continue to show the marked records, if any.

iv. Marked records

This command is only active when you are displaying search results. It allows you to view or edit only the records marked.

v. Open dictionary

Open the **dictionary window** of the active data base.

vi. Split/unsplit view

Split the data base window into two parts: the first displaying the record (according to the currently selected format) and the second part displaying either the current format or the current record in ASCII. If the window is already split when you select this command it will be returned to its normal view.

Selecting commands from this menu will modify the behavior of the navigation buttons in the **data base window** and the **data entry window** as follows:

	⏮	◀	▶	⏭
Data base	First data base record	Previous record	Next record	Last data base record
Search results **All records**	First record retrieved	Previous record retrieved	Next record retrieved	Last record retrieved
Marked records	First record marked	Previous record marked	Next record marked	Last record marked

vii. Print current record

Access a fast print dialog box for the current record. It is possible to print to the default printer as well as to an ASCII file. It is also possible to choose a different print format.

viii. Zoom (+) (-)

Increase or decrease the size of the content of the data base window.

D. Search menu

This menu contains the following commands:

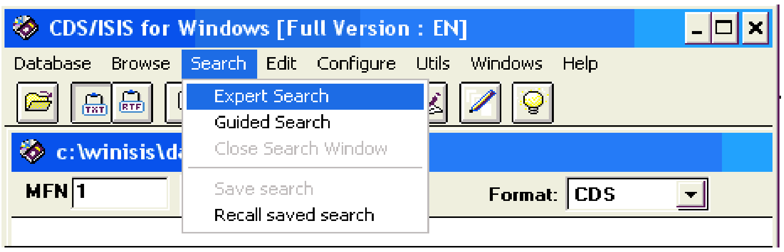


Figure 11 - Search Menu

i. Expert search

This command opens the standard **search window**, which allows you to search the data base. A simpler search method is provided by selecting the ***Guided Search*** command below.

ii. Guided Search

This command opens the **guided search window**, which provides a simplified search interface. Expert users may prefer to use the ***Expert Search*** command above.

iii. Close Search Window

This command closes the search window, if any, and activates the **data base window**.

iv. Save search

This command allows you to save the results of a particular search for later processing. When you select this command CDS/ISIS will first display the **search history** dialog box, from which you select the search to be saved, and then an **open dialog box**, where you can provide the name of the save file.

v. Recall saved search

This command allows you to recall a previously saved search (i.e. a search you have saved by means of the ***Save search*** command described above). CDS/ISIS will display an **open dialog box**, where you can select the save file to be recalled. The search corresponding to the recalled save file is then automatically added to the list of current searches and the **data base window** will display the first record retrieved. The recalled search becomes therefore the current search.

E. Edit menu

This menu contains the following commands:

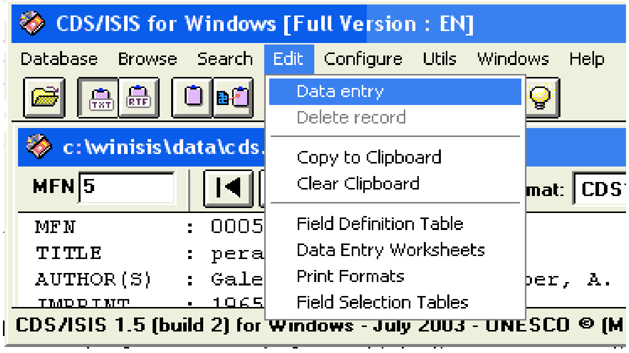


Figure 12 - Edit Menu

i. Data entry

This command opens the **data entry window**, which allows you to modify data base records.

ii. Delete record

While in data entry mode, this command deletes the current record (i.e. the record currently displayed in the data entry window).

iii. Copy to clipboard

This command copies the current record to the Windows Clipboard. It is only operational when the data base window is active. The record is copied in either ASCII or RTF. format depending on the currently selected option on the tool bar:

TXT Record is copied in ASCII format

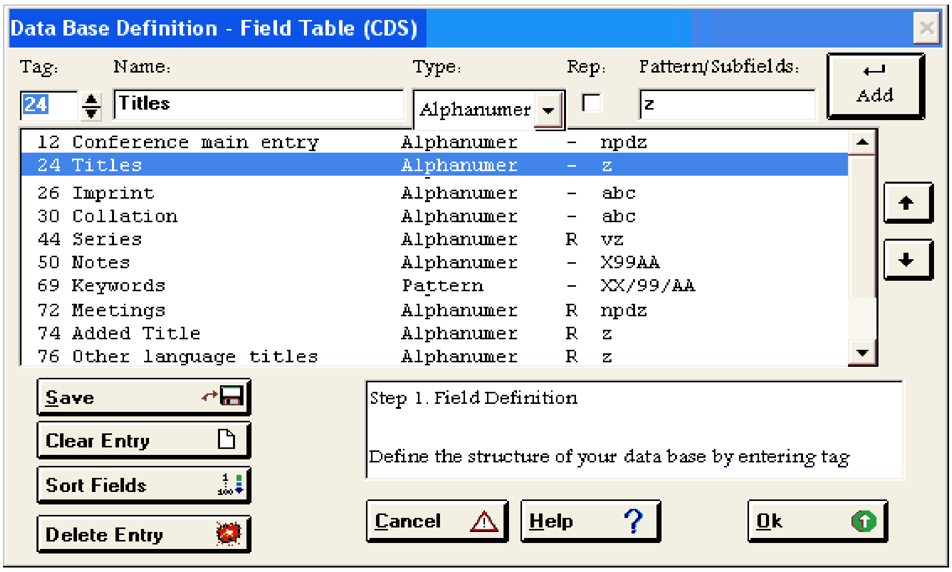
RTF Record is copied in RTF format

iv. Clear clipboard

This command clears the Windows Clipboard.

v. Field Definition Table

This command access Winisis' FDT editor, where it is possible to view or change the current database field definition:



Tag: 24 Name: Titles Type: Alphanumer Rep: □ Pattern/Subfields: z

Tag	Name	Type	Rep	Pattern/Subfields
12	Conference main entry	Alphanumer	-	npdz
24	Titles	Alphanumer	-	z
26	Imprint	Alphanumer	-	abc
30	Collation	Alphanumer	-	abc
44	Series	Alphanumer	R	vz
50	Notes	Alphanumer	-	X99AA
69	Keywords	Pattern	-	XX/99/AA
72	Meetings	Alphanumer	R	npdz
74	Added Title	Alphanumer	R	z
76	Other language titles	Alphanumer	R	z

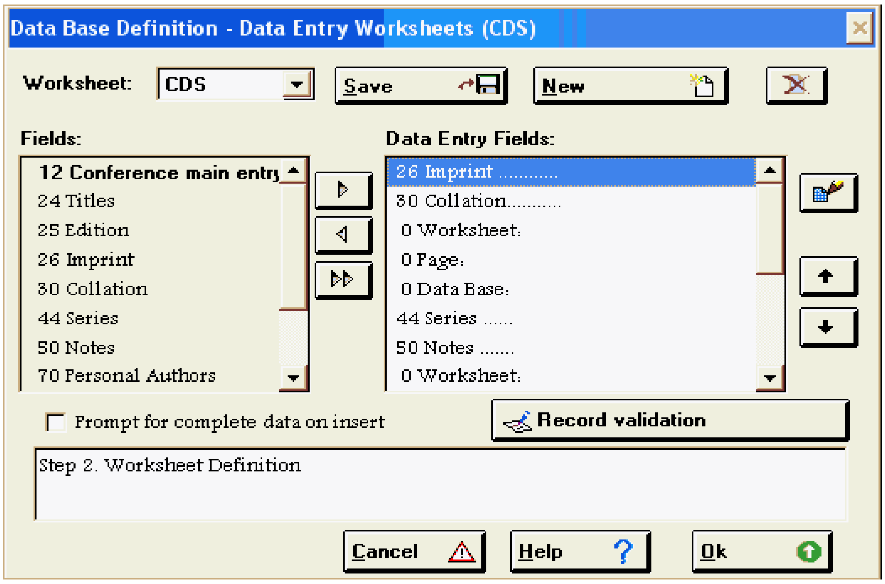
Step 1. Field Definition

Define the structure of your data base by entering tag

Figure 13 – The FDT Editor

vi. Data Entry worksheets

This command opens the data entry worksheets editor.



Worksheet: CDS

Fields:
12 Conference main entry
24 Titles
25 Edition
26 Imprint
30 Collation
44 Series
50 Notes
70 Personal Authors

Data Entry Fields:
26 Imprint ....
30 Collation..........
0 Worksheet:
0 Page:
0 Data Base:
44 Series ......
50 Notes ....
0 Worksheet:

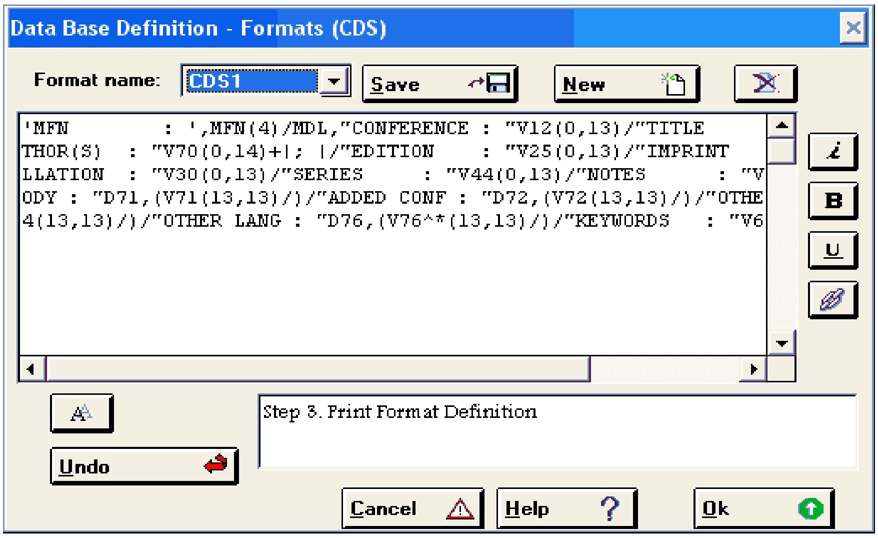
□ Prompt for complete data on insert

Step 2. Worksheet Definition

Figure 14 - FMT Editor

vii. Print Formats

This command opens the print format editor. It is an alternative solution to the editor integrated in the data base window. However, the new format cannot be directly tested.


Data Base Definition - Formats (CDS)

Format name: CDS1

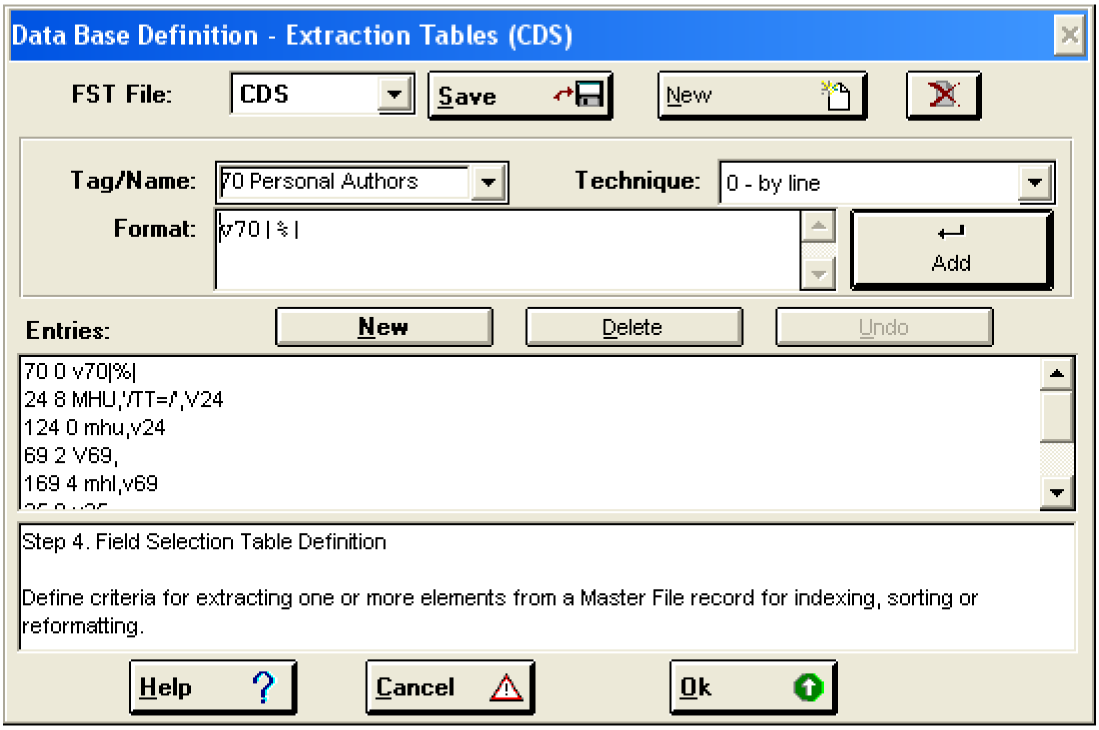
```
'MFN          : ',MFN(4)/MDL,"CONFERENCE : "V12(0,13)/"TITLE
THOR(S)   : "V70(0,14)+|; |/"EDITION    : "V25(0,13)/"IMPRINT
LLATION   : "V30(0,13)/"SERIES      : "V44(0,13)/"NOTES       : "V
ODY : "D71,(V71(13,13)/)/"ADDED CONF : "D72,(V72(13,13)/)/"OTHE
4(13,13)/)/"OTHER LANG : "D76,(V76^*(13,13)/)/"KEYWORDS   : "V6
```

Step 3. Print Format Definition

Figure 15 - PFT Editor

viii. Field Selection Tables

This command opens the internal FST Editor. FST files define the way the content of the data base is indexed and searchable. These rules, built up using the CDS/ISIS print formatting language, extract the searchable terms of the inverted file (dictionary of terms).


Data Base Definition - Extraction Tables (CDS)

FST File: CDS

Tag/Name: 70 Personal Authors

Technique: 0 - by line

Format: v70|%|

Entries:

```
70 0 v70|%|
24 8 MHU,'/TT=/',V24
124 0 mhu,v24
69 2 V69,
169 4 mhl,v69
```

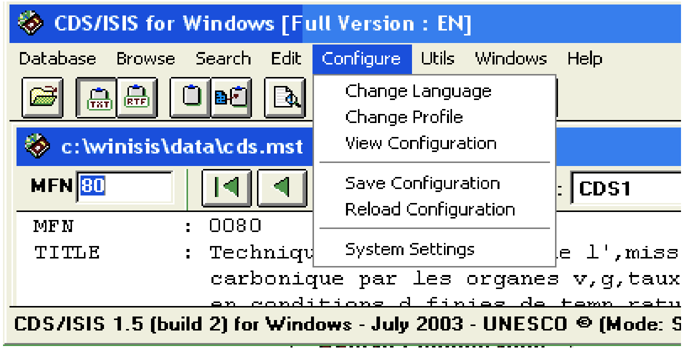
Step 4. Field Selection Table Definition

Define criteria for extracting one or more elements from a Master File record for indexing, sorting or reformatting.

Figure 16 - FST Editor

F. Configure menu

This menu contains the following commands:



```
MFN        :  0080
TITLE      :  Techniqu                    e l',miss
              carbonique par les organes v,g,taux
```

CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: S

Figure 17 - Configure Menu

i. Change Language

This command allows you to change the language in which menus, windows, prompts and system messages will be displayed. Initially CDS/ISIS will start in the language selected as default in parameter 101 of the **syspar.par** file. When you select this command a list of available languages is displayed from which you may select the one desired. Once selected, a language remains in effect until you change it again. Note, however, that any window open before you change the dialogue language, will remain in the language it was when it was initially opened.

ii. Change Profile

This command allows you to dynamically change certain system parameters.

iii. View Configuration

This command displays the current parameter settings.

iv. Save Configuration

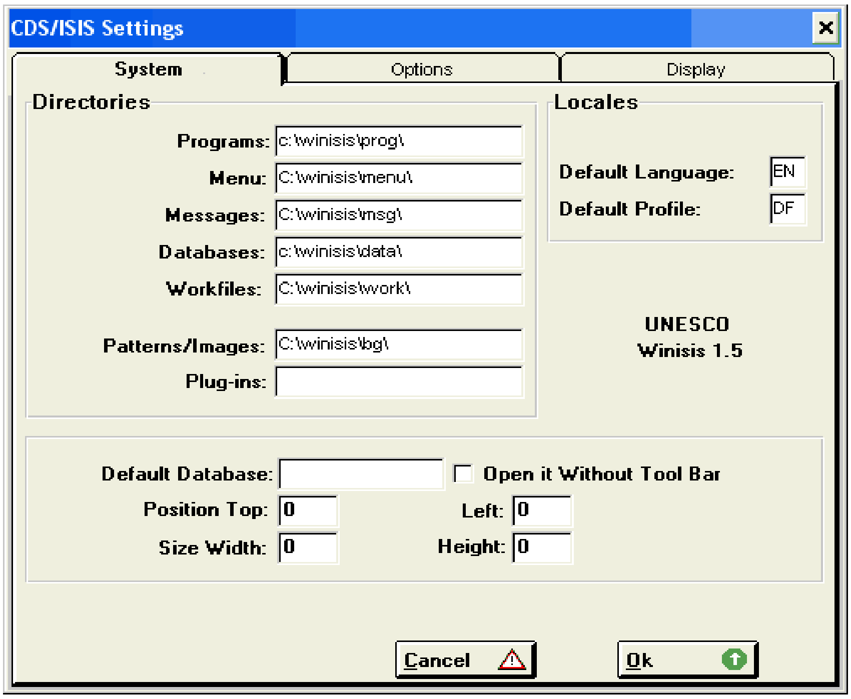
This command saves the current configuration in **syspar.par**. It updates, in particular, the following parameters: 101, 109, 111, 112 and 122.

v. Reload Configuration

This command reloads **syspar.par** from disk. If this file was edited since CDS/ISIS was started, the changed parameters will take effect immediately.

vi. System settings

Opens the “System settings” dialog box, which allows to change a number of Syspar.par parameters. It consists of three screens: the first one concerns pathnames, language, profiles and data base window position and size.



Directories

Programs: c:\winisis\prog\
Menu: C:\winisis\menu\
Messages: C:\winisis\msg\
Databases: c:\winisis\data\
Workfiles: C:\winisis\work\
Patterns/Images: C:\winisis\bg\
Plug-ins:

Locales

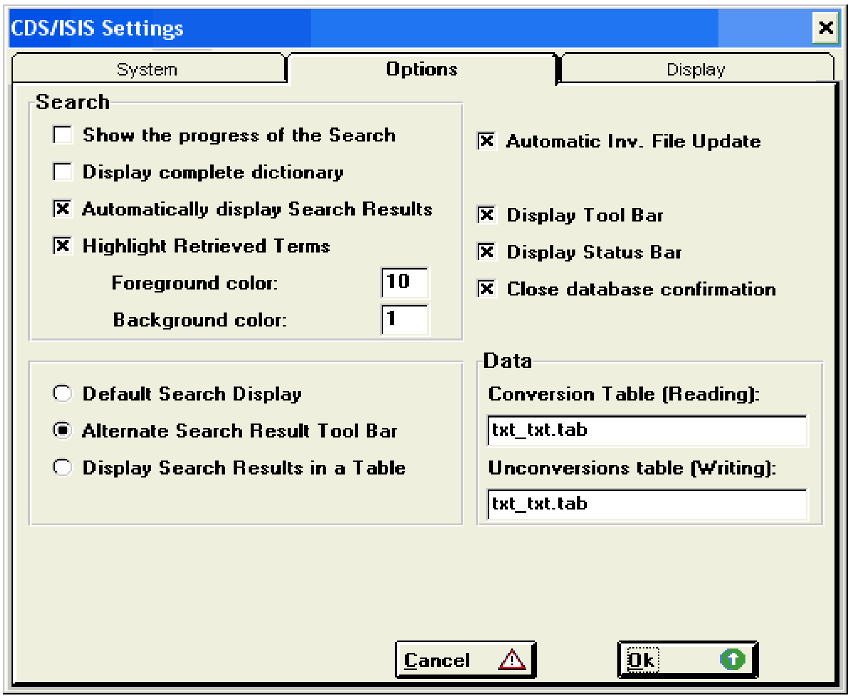
Default Language: EN
Default Profile: DF

UNESCO
Winisis 1.5

Default Database: □ Open it Without Tool Bar
Position Top: 0 Left: 0
Size Width: 0 Height: 0

Figure 18 - System settings screen 1

The second screen concerns settings affecting the behaviour of Winisis:



Search

□ Show the progress of the Search
□ Display complete dictionary
☒ Automatically display Search Results
☒ Highlight Retrieved Terms
Foreground color: 10
Background color: 1

☒ Automatic Inv. File Update
☒ Display Tool Bar
☒ Display Status Bar
☒ Close database confirmation

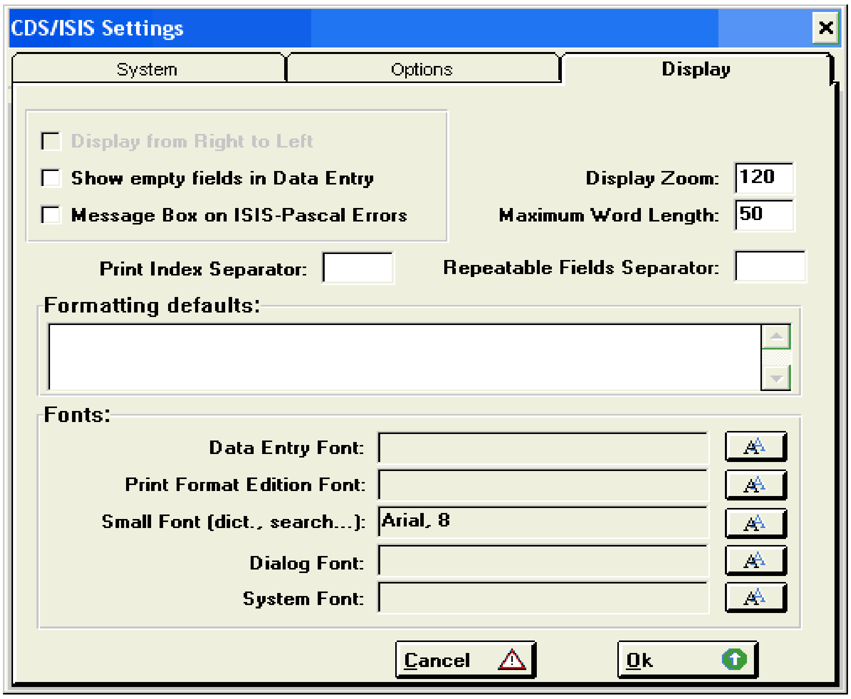
○ Default Search Display
● Alternate Search Result Tool Bar
○ Display Search Results in a Table

Data

Conversion Table (Reading): txt_txt.tab
Unconversions table (Writing): txt_txt.tab

Figure 19 - System settings screen 2

The third screen concerns mainly display options, such as font and zoom.



□ Display from Right to Left
□ Show empty fields in Data Entry
□ Message Box on ISIS-Pascal Errors

Display Zoom: 120
Maximum Word Length: 50
Print Index Separator:
Repeatable Fields Separator:

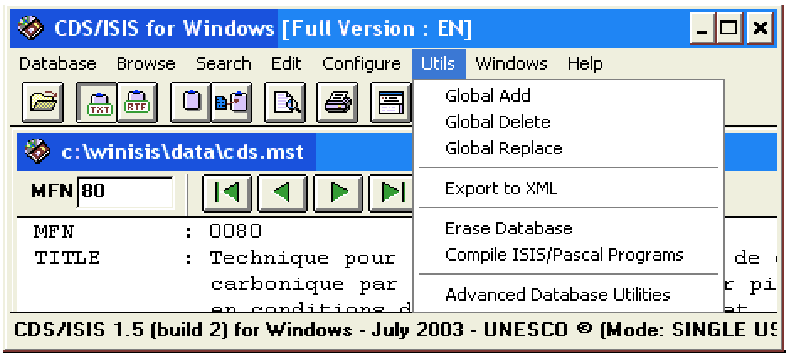
Formatting defaults:

Fonts:
Data Entry Font:
Print Format Edition Font:
Small Font (dict., search...): Arial, 8
Dialog Font:
System Font:

Figure 20 - System settings screen 3

G. Utilities menu

This menu contains the following commands:



```
MFN      : 0080
TITLE    : Technique pour
           carbonique par
```

CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: SINGLE US

Figure 21 - Utilities Menu

i. Global Add

This command allows you to add a field with a specified contents to a range of Master file records. CDS/ISIS will display the **Global Add Dialog box**.

ii. Global Delete

This command allows you to delete a field from a range of Master file records. CDS/ISIS will display the **Global Delete Dialog Box**.

iii. Global Replace

This command allows you to perform global changes in the data base (e.g. replace a string by another). CDS/ISIS will display the **Global Replace Dialog Box**.

iv. Export to XML

This command opens the Export to XML window, which allows defining a wide range of XML parameters, including the selection of which fields will be treated.

v. Erase database

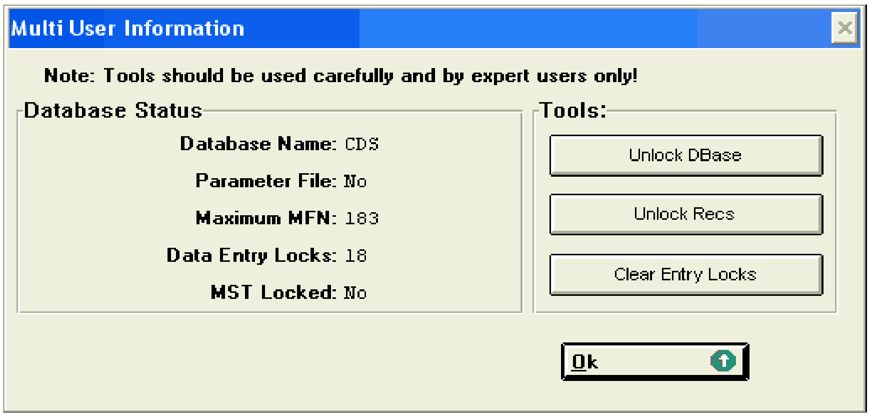
Caution: This command ***completely ERASES*** the content of the current Master File. It is **not** possible to undo this operation.

vi. Compile ISIS/Pascal Programs

This command allows you to compile a CDS/ISIS Pascal program. CDS/ISIS will first display an **open dialog box**, where you select the program to be compiled. If the program contains syntax errors, you will be allowed to edit it and compile it again. The compiler works in DOS window.

vii. Advanced database utilities

This command is mostly useful for CDS/ISIS network maintenance. It opens the following dialog box:



Multi User Information

Note: Tools should be used carefully and by expert users only!

Database Status

Database Name: CDS

Parameter File: No

Maximum MFN: 183

Data Entry Locks: 18

MST Locked: No

Figure 22 - Advanced data base utilities

On the left side there is some information on the status of the data base, taken from the MST Control record:

- **Database name**: active data base name
- **Parameter File**: “Yes” or “No” if the data base has been opened through a dbnpar.par
- **Maximum MFN**: the highest MFN number in the data base
- **Data Entry Locks**: how many users/sessions are actually working on the current database. This value may also be wrong and prevent the access to some delicate functions such as the “Inverted File Maintenance”. See “*Unlock Recs*” and “*Clear Entry Locks*”.
- **MST Locked**: “Yes” or “No” the Master file is locked. See “*Unlock DBase*”.

On the right side are the following tools:

CAUTION*: The use of these tools during active network sessions by more than one user, may cause data loss or damage.*

- **Unlock Dbase** clear the lock data base flag in the MST control record.
- **Unlock Recs** given a range of records, this tool will clear each record’s lock flag.
- **Clear Entry locks** clear the data entry lock counter in the MST control record.

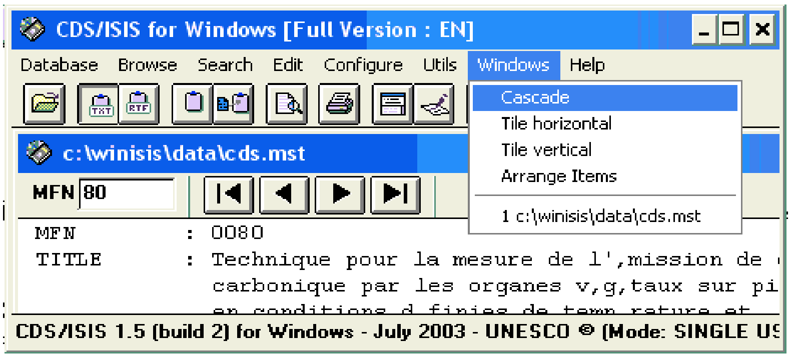
1. More hidden utilities

The following utilities can be added by editing the Menu file corresponding to your language:

- `Print dictionary[10145]`
  Open the Inverted file dump dialog.
- `Import from a HIT file[12912]`
  Allows importing the content of a HIT file into a data base. A HIT file, issued from a SORTED print run, contains the sort keys and headings.
- `Show registered plug-in list[10170]`
  Display the list of currently registered plug-ins. A plug-in becomes registered only after a CALL command.

H. Windows menu

This menu contains the following commands:



```
MFN        :  0080
TITLE      :  Technique pour la mesure de l',mission de
              carbonique par les organes v,g,taux sur pi
              en conditions d finies de temp rature et
```

CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: SINGLE US

Figure 23 - Windows Menu

i. Cascade

Arranges windows so that they overlap, starting in the upper-left corner of the main CDS/ISIS window. The title bar of each window remains visible, making it easy to select any window by simply clicking on the corresponding title bar.

ii. Tile

Arrange the windows side by side, so that each window is visible and no windows overlap.

iii. Arrange items

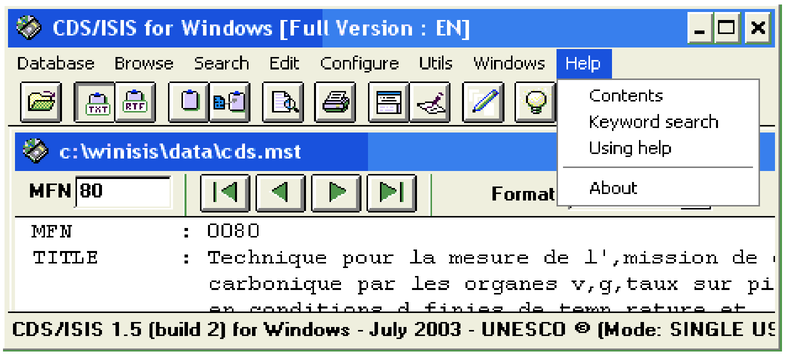
Arrange all icons in rows.

iv. 1, 2, 3, . . .

Select a particular window in the list of currently open data bases. CDS/ISIS changes this part of the Windows menu dynamically, as you open different windows. For each data base you may have up to three windows listed: the data base window, the search (or guided search window) and the data entry window. Depending on the number of windows open and the size of your screen, some may be hidden behind others. Selecting the desired window from this list will bring it to the foreground and activate it.

I. Help menu

This menu allows you to display the CDS/ISIS Help file and contains the following commands:



```
MFN        : 0080
TITLE      : Technique pour la mesure de l',mission de
             carbonique par les organes v,g,taux sur pi
             en conditions d finies de temp rature et
```

CDS/ISIS 1.5 (build 2) for Windows - July 2003 - UNESCO © (Mode: SINGLE US

Figure 24 - Help menu

Note: Winisis 1.5 HELP files are HTML based. They are NOT included in the current release but will be available separately on UNESCO's CDS/ISIS web page.

i. Contents, Keyword search and Using Help

These commands are all linked to the CDS/ISIS Help table of contents.

ii. About

This command displays the CDS/ISIS Copyright notice and the version number. This window also shows the unique ID number of the current installation. For example:

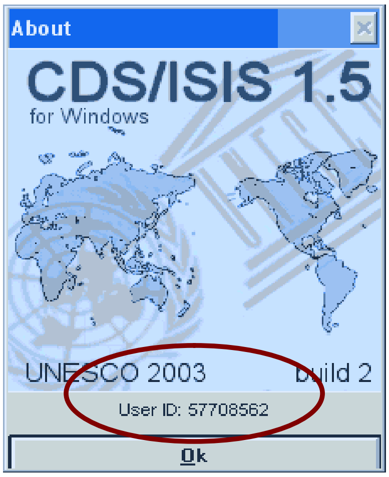
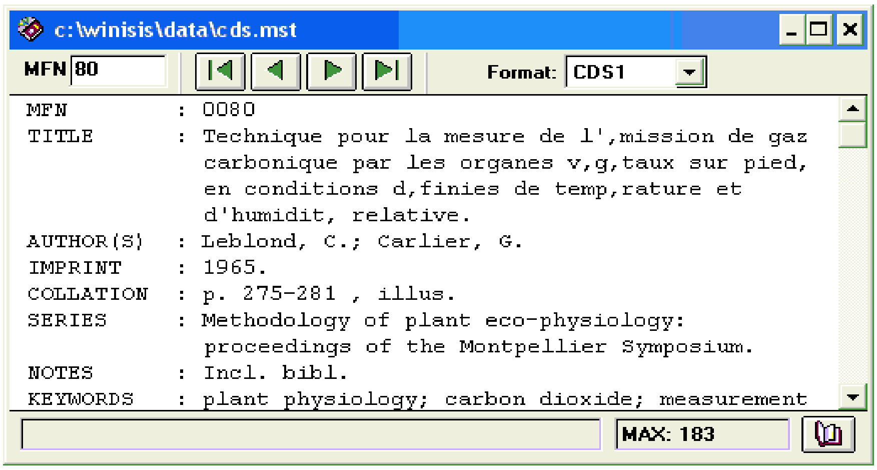


Figure 25 - About window and ID number

4. CDS/ISIS Windows

A. The Database Window

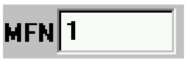
The data base window, a sample of which is shown below, is displayed when you open a data base. It displays the current record according to the currently selected display format. The contents of this window may be modified by selecting applicable commands in the **Browse menu**.



```
MFN         : 0080
TITLE       : Technique pour la mesure de l',mission de gaz
              carbonique par les organes v,g,taux sur pied,
              en conditions d,finies de temp,rature et
              d'humidit, relative.
AUTHOR(S)   : Leblond, C.; Carlier, G.
IMPRINT     : 1965.
COLLATION   : p. 275-281 , illus.
SERIES      : Methodology of plant eco-physiology:
              proceedings of the Montpellier Symposium.
NOTES       : Incl. bibl.
KEYWORDS    : plant physiology; carbon dioxide; measurement
```

Figure 26 - Data base window

The data base window toolbar contains the following items:



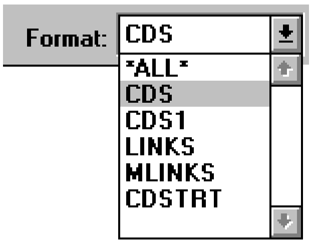
This field contains the current MFN number. Clicking on this field allows you to display a particular record by typing the desired MFN number and then pressing the Enter key.

Displays the first record. If you are displaying a search result the first record matching the search expression is displayed. If you are browsing the data base sequentially, the first data base record is displayed.

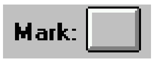
Displays the previous record. If you are displaying a search result the previous record (if any) matching the search expression is displayed.

Displays the next record. If you are displaying a search result the next record (if any) matching the search expression is displayed.

Displays the last record. If you are displaying a search result, the last record matching the search expression is displayed. If you are browsing the data base sequentially, the last data base record is displayed.



This field shows which format is currently in use. You can change the display format by first clicking on the field, which will pull down the list of display formats available, and then selecting a new display format. (The list of display formats is read from the FDT of the data base).



This feature is *available only when displaying a search* result. By pressing this button you can mark or unmark the current record (a "V" appears in the button when marked). After marking, you can display the currently marked records by choosing the Marked Records command from the Browse menu.

The status bar, at the bottom of the window, contains the following two fields:
Record Status: this field displays the status of the current record. The status is not displayed when the record is in a normal state, otherwise it indicates that the record has been deleted. When displaying search results, a string like the following is displayed:

Search #5: [1/10]

In this example you are displaying the first of the 10 records found in the 5th search expression you have submitted.
Maximum MFN: displays the maximum master file number in the current database.

When a data base window is active you may also use the following buttons of the main window tool bar:

This button opens another data base. It has the same effect as selecting the **Open** command in the **Database** menu.

Clears the Windows Clipboard. It has the same effect as selecting the **Clear Clipboard** command in the **Edit** menu.

Copies the record being displayed to the Windows Clipboard. It has the same effect as the **Copy to Clipboard** command in the **Edit** menu. Note that you can concatenate several records. If RTF mode is set (see below), you will obtain a single RTF document containing all the records you have copied into the Clipboard.

Sets the Clipboard mode to ASCII format.

Sets the Clipboard mode to RTF format.

Opens the Search window for the current data base. It has the same effect as the search commands in the **Search** menu. Normally the type of search window which is activated is the same as the last Search menu command you have used, i.e. **Expert search** or **Guided search**.

Starts a print run. It is equivalent to the **Print** command in the **Database** menu. You may print the output of a search and/or a selected range of records and you may sort the records by any combination of fields and subfields. The printing parameters are specified in the **Print Dialog Box**.

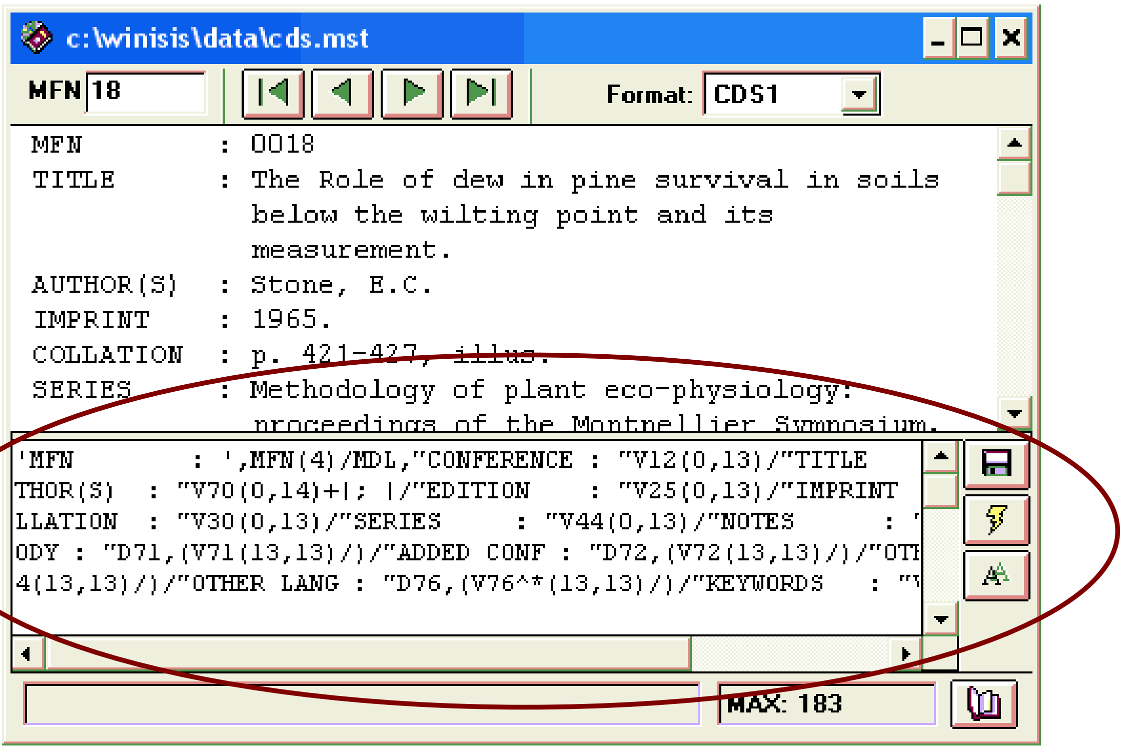
Splits the data base window into two parts: the first displaying the record in graphic mode and the second one displaying the record in ASCII format. If the window is already split, it is restored to its normal (unsplit) status.

Splits the data base window into two parts. In a split window this button toggles the second part of the window to display either the current format or an ASCII view of the record. If the window is not already split, it splits it and displays the current format.

Opens the Data Entry Window for the current data base. It has the same effect as the **Data entry** command in the **Edit** menu.

Opens the data base definition Wizard, a guided environment for creating new databases.

Using toolbar button “**Split**” (), the data base window will show the current display format:



```
MFN          : 0018
TITLE        : The Role of dew in pine survival in soils
               below the wilting point and its
               measurement.
AUTHOR(S)    : Stone, E.C.
IMPRINT      : 1965.
COLLATION    : p. 421-427, illus.
SERIES       : Methodology of plant eco-physiology:
               proceedings of the Montpellier Symposium.
```

```
'MFN        : ',MFN(4)/MDL,"CONFERENCE : "V12(0,13)/"TITLE
THOR(S)   : "V70(0,14)+|; |/"EDITION    : "V25(0,13)/"IMPRINT
LLATION   : "V30(0,13)/"SERIES     : "V44(0,13)/"NOTES      : '
ODY : "D71,(V71(13,13)/)/"ADDED CONF : "D72,(V72(13,13)/)/"OTH
4(13,13)/)/"OTHER LANG : "D76,(V76^*(13,13)/)/"KEYWORDS   : "V
```

Figure 27 - Data base window built-in format editor

The user will be able to change, try new commands and save changes to the disk using the following buttons:

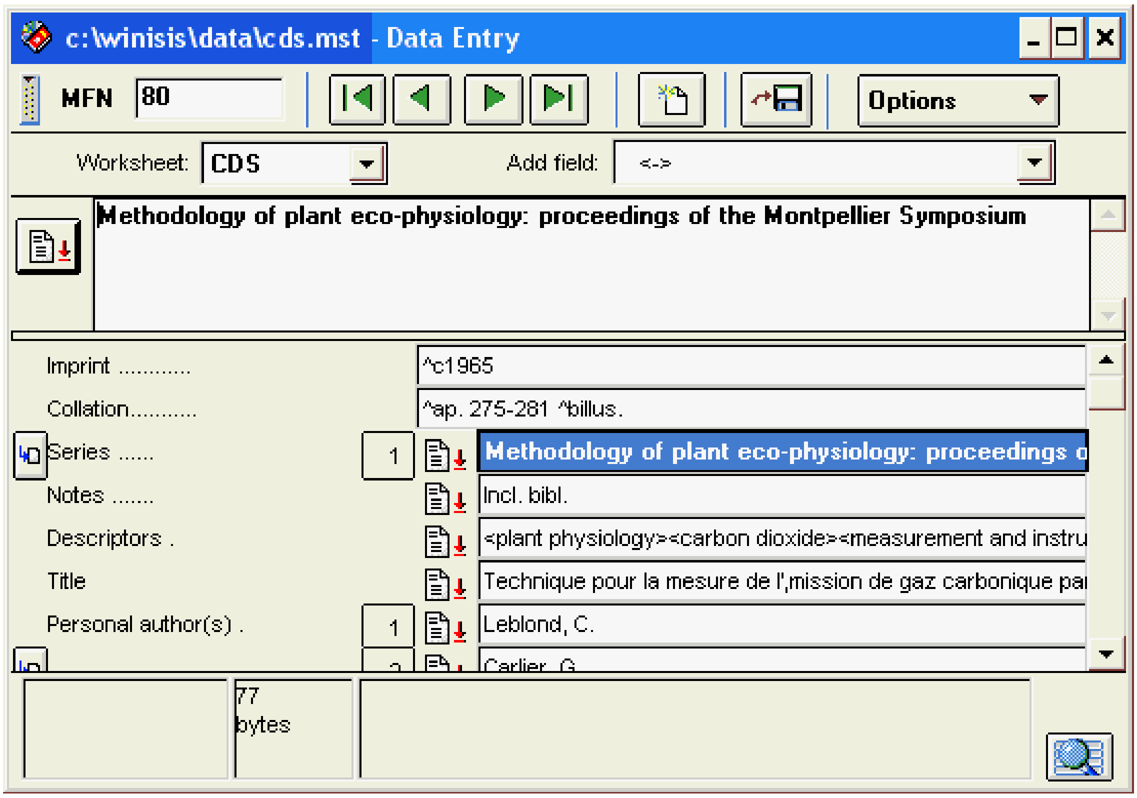
Save format to the disk

Execute the format as it is. Note that this action does not replace the existing format and that, unless you save it, it will be always possible to return it to its original format.

Change the font used to edit formats. This option is particularly useful in order to be able to see all format control characters that may not be visible in all fonts.

B. The data entry window

This window is displayed whenever you select the ***Data Entry*** command of the **Edit** menu.



c:\winisis\data\cds.mst - Data Entry

MFN 80

Worksheet: CDS

Add field: <->

Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

Imprint .... ^c1965

Collation........... ^ap. 275-281 ^billus.

Series ...... 1 Methodology of plant eco-physiology: proceedings o

Notes .... Incl. bibl.

Descriptors . <plant physiology><carbon dioxide><measurement and instru

Title Technique pour la mesure de l',mission de gaz carbonique pai

Personal author(s) . 1 Leblond, C.

2 Carlier, G.

77 bytes

Figure 28 - The Data Entry Window

When initially opened, this window displays the current record (i.e. the one which is currently displayed in the corresponding data base window). Its contents, however, may be modified by selecting applicable commands in the **Browse menu**.

The fields of the record are displayed in the lower part of the window according to the current (or default) data entry worksheet.

The data entry window toolbar contains the following items:

This toggle switch allows you to show or remove the second toolbar.

MFN 1 This field contains the current MFN number. Clicking on this field allows you to edit a particular record by typing the desired MFN number and then pressing the Enter key.

Displays the first record. If you are editing a search result the first record matching the search expression is displayed. If you are editing the data base sequentially, the first data base record is displayed.

Displays the previous record. If you are editing a search result the previous record (if any) matching the search expression is displayed.

Displays the next record. If you are editing a search result the next record (if any) matching the search expression is displayed.

Displays the last record. If you are editing a search result the last record matching the search expression is displayed. If you are editing the data base sequentially, the last data base record is displayed.

Creates a new record. The current worksheet is displayed with all its fields empty.

Saves the current record in the Master file.

This toggle switch allows you to show (or remove) empty fields from the screen (normally when a record is initially displayed, empty fields are not automatically shown, unless parameter 123 of syspar.par is set to 1).

Worksheet: CDS

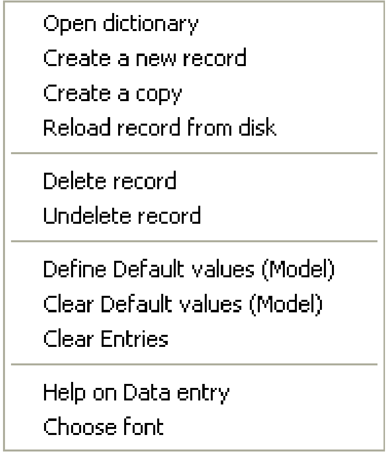
Allows you to select a different worksheet. By clicking on this field the list of available worksheets (as defined in the FDT) is displayed.

Add field: <->

Allows you to add a new field in the record. By clicking here the list of fields in the current worksheet is displayed.

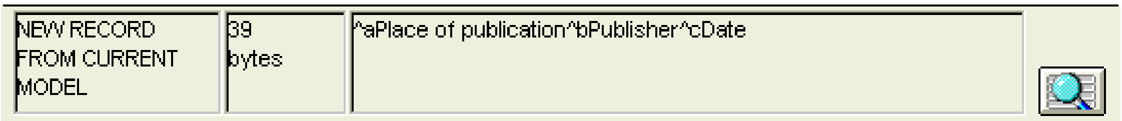
Options

Additional options are available by pressing this button. When pressed the following sub-menu is displayed:



Open dictionary: Opens the Dictionary window.
Create new record: Creates a new record. The current worksheet is displayed with all its fields empty.
Create a copy: Creates a new record with the same contents of the current one. The created record is assigned the next available MFN.
Reload record from disk: Cancels all the changes made and restores the record to its initial status.
Delete record: Marks the current record as (logically) deleted.
Undelete record: Undeletes a (logically) deleted record.
Define Default values (Model): sets the current record as a record template. A new record is created (but not stored on the disk) using that content.
Clear Default values (Model): Clear default values. New records will be empty.
Clear entries: Clears the contents of all the fields in the worksheet. You may use this option to replace an existing record with a new one having the same MFN. Note, however, that only the fields present in the worksheet are cleared. Other fields present in the record but not in the worksheet retain their current value.
Help on data entry: Displays the data entry topic in the CDS/ISIS Help.
Choose font: Allows you to change the font and font size used for the Field edit box.

The status bar, at the bottom of the window, contains the following three fields:



NEW RECORD FROM CURRENT MODEL

39 bytes

^aPlace of publication^bPublisher^cDate

Figure 29 - Data entry Status Bar

Record Status: this field displays the status of the current record. The status is not displayed if everything is normal, otherwise it indicates that the record has been deleted or if a model is currently set. When editing search results, a message is displayed, for example:

Search #5: [1/10]

This states that you are editing the first of the 10 records found in the 5th search expression you have submitted.

Bytes: the length of the current field in bytes (characters).

Help message: displays the help message, if any, associated with the field being edited.

1. Editing a field

To edit a particular field you must first click on the corresponding field button. Its contents will then be brought up in the ***field edit box***. In editing a field you may use all standard Windows edit function keys (see Appendix A) and, in addition, the three following special keys:

Enter	update the field and select the next field;
Shift+Enter	update the field and select the previous field;
F2	***delete the field***;
ESC	ignores all changes (field will redisplay as it was before the editing started).

You may change the size of this box by dragging up or down the bar at the bottom of the box. You may also change the font used to display the field by selecting the ***Choose font*** option from the **Options** sub-menu.

a. Subfielded fields

When you enter a field containing subfields you must key in the required subfield delimiters in front of each subfield. A subfield delimiter is a 2-character code preceding and identifying a variable length subfield within a field. It consists of the character ^ followed by an alphabetic or numeric character, e.g. ^**a**.

If the subfield code is alphabetic, you may enter it in either upper or lower case: CDS/ISIS makes no difference between ^**a** and ^**A**. You may therefore use the most convenient form.

Do not insert spaces or punctuation marks either before or after the subfield delimiter, unless you have been specifically instructed to do so. Entering spaces or punctuation may adversely affect the printing of the field later on.

Here is an example of a field with three subfields:

```
^aUnesco^bParis^c1985
```

b. Repeatable fields

If the field you are entering is repeatable and you need to enter more than one occurrence, enter each occurrence separately, and click on the repeatable field icon (preceding the field name) for each new occurrence to be added. Alternatively, you may add as many occurrences as needed in the Field edit box, separating each by a percent (**%**) sign[4]. In this case you should not enter spaces around the % sign. If you do they will be entered in the field. The example below shows an author field with two names:

```
Brown, J.%Johnson, Archibald
```

You may use this technique whenever you want to insert a new occurrence between two existing ones.

[4] Note that the actual character used to separate occurrences of a repeatable field is an installation-dependent parameter (see Parameter 8 of the syspar.par file in the Section “System Parameter Files”). If this has been re-defined, you must of course use the character assigned to this purpose rather than the % sign.

c. Control characters

Certain characters stored in a field, although keyed in as data, will be interpreted by CDS/ISIS as ***control characters***, rather than data characters, and will normally activate some special type of processing. Control characters are normally reserved for CDS/ISIS use and may not therefore be used as data.

Subfield delimiters are an example of control characters. Other control characters recognised by CDS/ISIS are described below.

i. Search term delimiters

Search term delimiters may be used to identify key terms or phrases assigned to each record to enable its retrieval. Keywords may be delimited in either of two ways: by enclosing them between a pair of slashes (/..../) or by enclosing them in triangular brackets (<....>). The advantage of using triangular brackets over using slashes, is that, these, unlike slashes, are reserved characters, and CDS/ISIS provides options to either display the brackets or suppress them, whereas *no option is provided to suppress slashes*.

When brackets are suppressed, they are normally deleted from the displayed version of the field, except when an open bracket immediately follows a closed one: in this case CDS/ISIS will replace them with a semicolon and a space. For example, by selecting the appropriate display mode the following entry:

```
<university course><documentation training><library school>
```

will be displayed as follows:

```
university course; documentation training; library school.
```

Except for the case mentioned above, you must ensure that the required spaces precede and follow the open and closed bracket respectively. For example when keywords are embedded within other text in the field as below:

```
Mission report describing a <university course> in <documentation
training> at an East African <library school>
```

the spaces surrounding the keywords *must* be present in order to produce the correct display:

```
Mission report describing a university course in documentation
training at an East African library school
```

If the field was entered as follows:

```
Mission report describing a<university course>in<documentation
training>at an East African<library school>
```

CDS/ISIS would display it as:

```
Mission report describing auniversity courseindocumentation
trainingat an East Africanlibrary school
```

In other words, CDS/ISIS simply ignores the brackets and ***does not* replace** them with spaces.

ii. Filing information

When producing printed catalogues you will need to sort the contents of one or more fields in order to print the records in the required sequence. CDS/ISIS will try to produce a sorting sequence according to normally accepted filing rules, but sometimes this will not be possible. In these cases CDS/ISIS offers you the possibility to state explicitly how a given field must be sorted by supplying ***filing information*** at the time you enter the data. Filing information is permanently recorded in the field.

This facility allows you to instruct CDS/ISIS to replace or ignore any sequence of data characters in a field whenever the field is used as a filing element, by using one of the following specifications:

<text-a=text-b> in this case, CDS/ISIS will replace **text-a** by **text-b** when the field is used in sorting, but use **text-a** (and ignore **text-b**) when displaying the field;

<text-a> in this case **text-a** will be ignored when sorting and only used to display the field.

Below are a few cases in which this facility is normally used (but its use is not restricted to just these cases):

Entered as	Sorted as	Displayed as
<The> evolution of information systems	EVOLUTION OF INFORMATION SYSTEMS	The evolution of information systems
<100=onehundred> days	ONEHUNDRED DAYS	100 days
<Mc=mac>Pherson, J.	MACPHERSON J.	McPherson, J.

Note that the use of MHL in sort keys will prevent the functioning of the above technique.

d. Inserting dictionary terms in a field

You may insert dictionary terms at the current cursor position of the edit box by pressing the ***Dictionary button*** which will open the **Dictionary Window**.

e. Inserting file names in a field

You may rapidly insert filenames into fields. To do this first open the Windows File Manager, then select the file(s) you want to insert in the current field and drag and drop them over the field edit box. In this way you may easily establish links between fields and external files such as pictures, movies, sounds, etc.

2. Adding a field

Select the field you want to add from the **Add field** list. Note that you may not add a second occurrence of a non-repeatable field. If one or more occurrences of a repeatable field already exist, a new occurrence will be added. Another way to add a new occurrence of a repeatable field is to click on the icon preceding the field name.

3. Deleting a field

First click on the field button, then press **F2** or delete the contents of the field and press the **Enter** key.

4. Field and record validation

CDS/ISIS validates each field you enter according to the field type defined in the FDT of the data base. For example It does not allow you to enter more than one occurrence of a non repeatable field, and it checks the validity of subfield identifiers.

In addition, you may attach a data entry validation file to each data base. This file, when present, must be in the data base directory and its name must be ***dbn*.val** (where ***dbn*** is the data base name).

The validation file is an ASCII file which you create using a text editor (e.g. Notepad). It consists of one or more lines, each associated with a particular field, and may include a line for global record validation. The field validation lines have the following format:

```
        tag: format
```

where **tag** is the field tag to be validated and **format** is a CDS/ISIS format. Each time a field is created and/or modified, CDS/ISIS will execute the corresponding validation format. Any output produced by the format is displayed as an error message.

The record validation line, which must always be the last line in the validation file, has the following format:

```
        :format
```

This format is executed immediately before updating the current record. Any output produced by the format is displayed as an error message.

Note that each field or record validation format *must* be contained in a single line. You must not therefore insert a carriage return other than at the end of the format.

A sample validation file is given below:

```
10: &val10()
20: if size(v20)>7 then 'Field 20 may not be more than 7 characters long' fi
:if p(v20) and p(v10) then 'Field 10 and 20 are mutually exclusive' fi
```

The above validation file validates fields 10 and 20 and the whole record. For field 10 it calls format exit val10. If val10 returns an non empty string, then the returned string is displayed as an error message. For field 20 an error message is displayed if the field is longer than 7 bytes. Finally, an error message is displayed if both fields 10 and 20 are present in the record which is about to be updated.
If a field or record does not match the validation criteria, CDS/ISIS issues an error message and repositions the cursor at the beginning of the field. You must then make whatever corrections are required before proceeding to the next field.

Note: *Each validation statement can be up to 8000 characters, but each VAL file line may be up to 1024 characters*

4. BEGIN and END coding

In certain cases it would be desirable to intervene on the record's content just before or after the user modifications. For that purpose, the VAL (validation) file contains two sections allowing executing a format that may change the content of the current record at the beginning of the editing and/or at its end. The format must output a character string compatible with **BIREME's CISIS Field Update** language specifications.

The following commands are available:

d. Makes the record logically deleted
d* Deletes all fields
dtt Deletes all occurrences of field tt
dtt/occ Deletes occurrence occ of field tt
att#str# Adds string str as a new occurrence of field tt (note that you may use any character as a delimiter instead of #)
htt n str_n Adds string *str_n* which is *n* bytes long as a new occurrence of field tt

Example of CDS.VAL file:

```
begin:if mfn=0 then 'a12#UNESCO#' fi
end:if IsChanged() then 'd2a2#',date(2),'#',fi
```

The first sentence can be used to give a dynamic default value if the record is **new** (MFN is zero). The second sentence can be used to assign a field with the **date** of the last modification. Remember that you can use parameter **150** to configure the output of function `date()`.

`mfn=0` will test if the current edited record is a new one (MFN number has not been assigned yet) or an old one.

The following **functions** are also available in this context (note that names are case sensitive):

IsChanged() is a Validation-specific function reporting if the record content has changed. Makes sense in END code only.

StoreMFN(*tag*) used on new records will store the newly assigned Master File Number to the specified field tag. Makes sense in END code only and exclusively on new records. *Warning: this function prevents the writing checking on new records.* Example:

```
If mfn=0 then StoreMFN(222) fi,
```

The above line will assign the new MFN number to field 222.
StoreMFN has **not** been tested intensively in network environments.

GetWorksheetName() returns the name of the current worksheet between single quotes. For example:

```
if GetWorksheetName()='CDS' then 'a69#<higher education>#',fi
```

The following **action** is also available in this context:

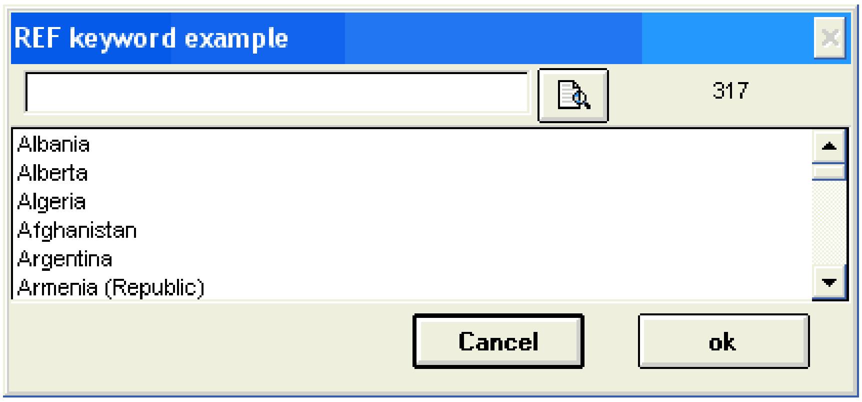
'WORKSHEET:xxxx'/ will **change the current worksheet** to xxx. It must be followed by a carriage return. For example:

```
if a(v12) then 'WORKSHEET:CDSA'/, else 'WORKSHEET:cds'/,fi
```

will change the current worksheet to `CDSA.FMT` if field 12 is absent or to `CDS.FMT` if not.

5. Pick lists

CDS/ISIS for Windows manages data entry pick lists, that can be defined using the VAL (validation) file:



Albania
Alberta
Algeria
Afghanistan
Argentina
Armenia (Republic)

Figure 30 - Pick list example

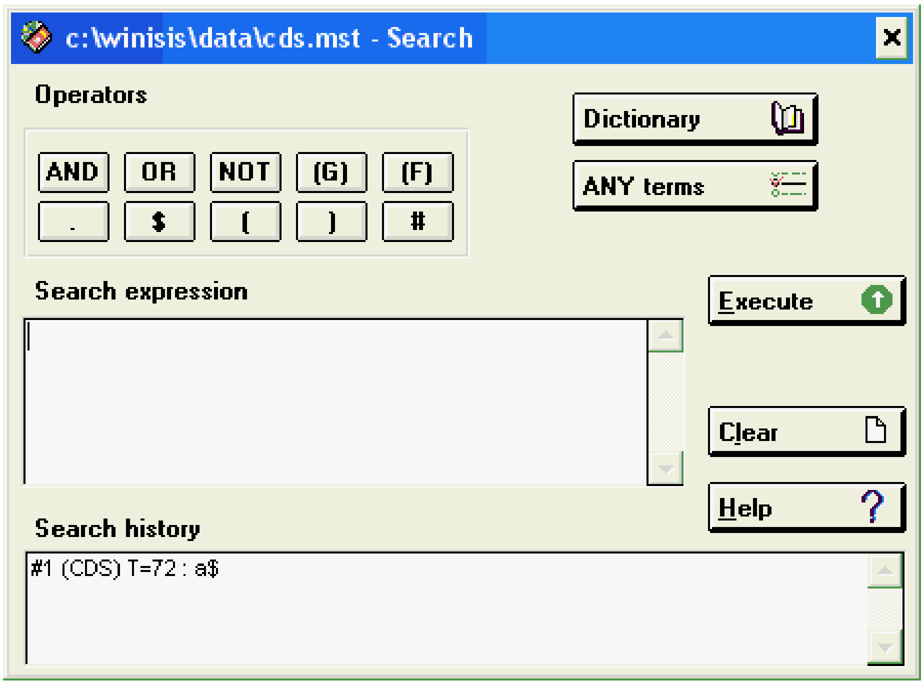
For more information on how to implement and use pick-lists, please refer to the Pick-lists HOW-TO document (separate).

6. Updating the Inverted file

You should be aware that CDS/ISIS may not automatically update the Inverted file whenever you add, modify or delete a Master file record (this depends on the settings of parameter 130 in the **syspar.par** file). It is important to remember this because, until such time as the Inverted file is updated, the following situation will exist:
new records you have added are not available for retrieval;
records you have modified are retrievable, but under the old access points; and
records you have deleted will still be posted to their access points, although the records themselves are no longer displayed.
However, CDS/ISIS keeps in the Master file all the information required to update the Inverted file and you may therefore request an Inverted file update at any time by selecting the ***I/F Update*** command of the **Data base** menu. This operation may take some time depending on the number of records updated, the computer you are using and the average number of search terms generated for each record. After some experience with a given data base, you will be able to estimate fairly closely the time required to update a record, and if you find that it is acceptable we advise you to always update the Inverted file after each data entry session (or before exiting the program).
Alternatively, you may defer the updating of the Inverted file to a more convenient time (e.g. at the end of the day, of the week, etc.).

C. The Expert Search Window

This window is displayed by selecting the ***Expert Search*** command in the **Search menu** (*see also:* The Guided Search window).



#1 (CDS) T=72 : a$

Figure 31 - Expert Search window

i. Operators

Pushing any of these buttons will insert the corresponding operator in the *Search expression* box.

ii. Search expression

This edit box contains the current search expression. Experienced users may type the search expression directly in this box (*see* the section "The CDS/ISIS Search Language" on p. 95 of the *CDS/ISIS Reference Manual*). The maximum number of characters you may type into this box is 1000.

The following buttons operate on this box:

DICTIONARY displays the **Dictionary window**.

ANY terms displays the list of ANY terms for the data base. Clicking on a term will insert it in the *Search expression* box.

EXECUTE pressing this button executes the search expression

CLEAR pressing this button erases the search expression

CANCEL pressing this button returns to the **data base window**

HELP pressing this button displays the search help topic

iii. Search history

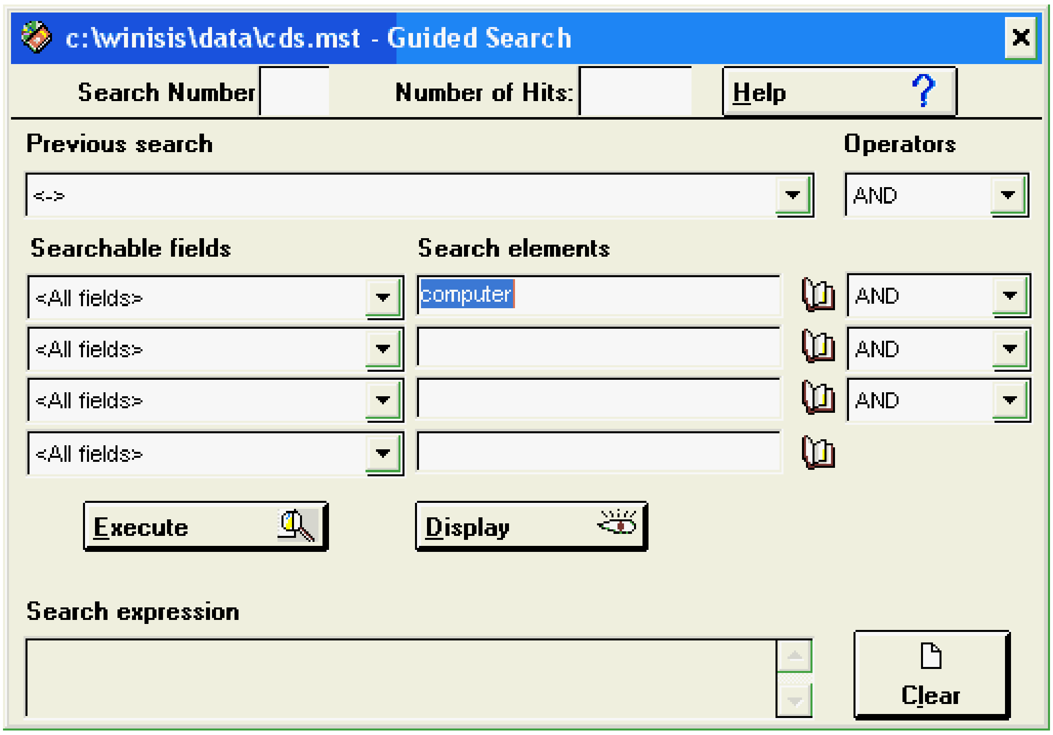
This box contains the list of the search expressions which have been executed so far. For each expression it gives the set number, the data base name, the number of hits and the search expression. A new element is added to this list each time a search expression is executed.

Double clicking an element of this list will display the corresponding results in the **Data base window**.

Single clicking usually inserts the set number in the ***Search expression*** box. However if the selected element corresponds to a search done on a different data base the search expression itself is copied. You may also copy a search expression (rather than the set number) to the ***Search expression*** box by dragging the element from the ***Search history*** list and dropping it in the *Search expression* box.

D. The Guided Search Window

This window provides a simplified search interface for inexperienced users. It is displayed by selecting the ***Guided Search*** command in the **Search** menu (*see also:* The Expert Search window and the section “The CDS/ISIS Search Language”).



Previous search: <->

Operators: AND

Searchable fields	Search elements	Operators
<All fields>	computer	AND
<All fields>		AND
<All fields>		AND
<All fields>		

Figure 32 - The Guided Search window

i. Search elements

Four text boxes are provided where you may type a search term. Alternatively you may press the ***Dictionary*** button (📖) to displays the **Dictionary window** (see below) from which you may then select the appropriate search term.

ii. Searchable fields

Each search element may be associated with a particular field by selecting the appropriate one from the corresponding ***Searchable fields*** list.

iii. Operators

Search elements may be linked with one of the three search operators AND, OR, or NOT, by selecting the appropriate one from the corresponding ***Operators*** list.

iv. Previous search

When one or more previous searches have been made, you may select one from the ***Previous search*** list and connect it with a new set of search elements.

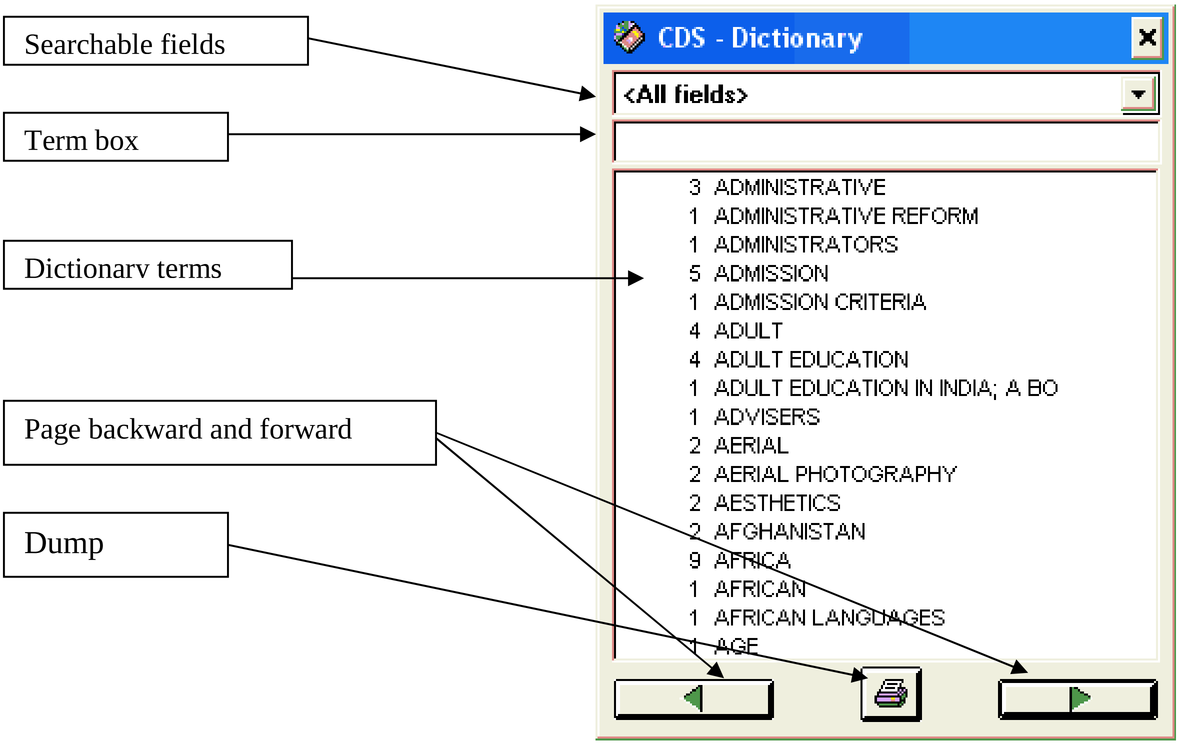
v. Other buttons

To execute the search press the ***Execute*** button. The actual text of the search statement generated by CDS/ISIS will then be displayed in the ***Search expression*** box. When the search is completed its number will be shown in the ***Search Number*** box and the total number of records retrieved in the ***Number of Hits*** box.

You may display the results of the search by pressing the ***Display*** button;. pressing the ***Cancel*** button returns to the data base window without executing the search whereas pressing the ***Clear*** button clears all the ***Search elements***.

E. The Dictionary Window

This window displays the dictionary of search terms. It is displayed whenever you press the Dictionary button on the **Expert Search** window, the **Guided Search** window, the **Data Entry** window or the **Data base** window.



```
3  ADMINISTRATIVE
1  ADMINISTRATIVE REFORM
1  ADMINISTRATORS
5  ADMISSION
1  ADMISSION CRITERIA
4  ADULT
4  ADULT EDUCATION
1  ADULT EDUCATION IN INDIA; A BO
1  ADVISERS
2  AERIAL
2  AERIAL PHOTOGRAPHY
2  AESTHETICS
2  AFGHANISTAN
9  AFRICA
1  AFRICAN
1  AFRICAN LANGUAGES
1  AGE
```

Figure 33 - The Dictionary window

i. Searchable fields

This combo box allows you to select the field to be searched when you are in expert search mode. Selecting a field will automatically append the corresponding **term qualifier** to search terms selected from the dictionary. Selecting **<All fields>** will search in every field.

When a field is selected, the Dictionary window is modified as follows, depending on the setting of parameter 104 in **syspar.par**.

104=0 (default) the full dictionary is displayed but terms in other fields are displayed in lower case and are not selectable;

104=1 only the dictionary terms in the selected field are displayed.

ii. Term box

Typing one or more characters in the ***Term*** box will position the dictionary window accordingly (e.g. typing F will show the dictionary terms starting with F).

iii. Dictionary terms

Double clicking on a Dictionary term will select it and, depending on the Window from which the Dictionary was called, insert it as follows:

Expert Search Window: the term will be inserted at the current cursor position in the ***Search expression*** box. Unless an operator has been previously selected an OR operator is automatically added if necessary;

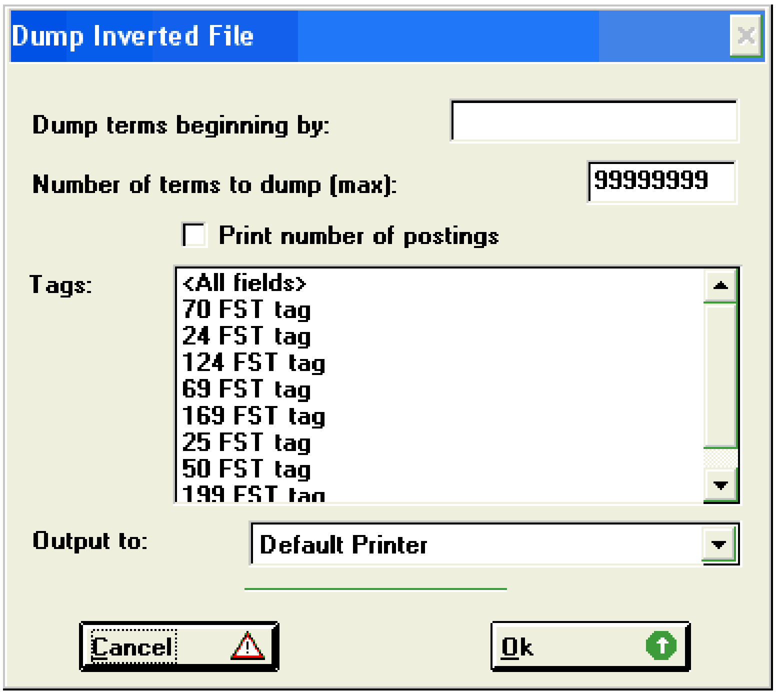
Guided Search Window: the term will be inserted in the current ***Search element*** box;
Data Entry Window: the term will be inserted at the current cursor position in the ***Field Edit*** box.
Note that the same operation may also be performed by dragging the term to the corresponding window. Use the method you find most convenient.

iv. Page forward and backward buttons

These buttons will page forward and backward. Note, however, that backward paging will go no further than the term from which a forward paging was started. To move to different parts of the dictionary use the **Term** box.

v. Dump dictionary button

This button opens the Dictionary Dump dialog box, which makes it possible to output the whole (or a part of) dictionary on a printer or a file.



Dump terms beginning by:

Number of terms to dump (max): 99999999

□ Print number of postings

Tags:
<All fields>
70 FST tag
24 FST tag
124 FST tag
69 FST tag
169 FST tag
25 FST tag
50 FST tag
199 FST tag

Output to: Default Printer

Figure 34 - Inverted File dump

vi. Modifying the behaviour of the Dictionary window

The dictionary window accepts three parameters in the database file EXP. The file EXP is usually located in the same folder as the PAR file (the data base default directory).
Those parameters are valid for both the **Expert** and the **Guided** search windows. Parameters are:

550=indicates which is the current prefix for a given field tag.
Syntax: `550=tag prefix (repeatable)`
For example:

550=24 TI=
550=70 AUT=

This allows Winisis to hide prefixes in the dictionary: when selecting a specific field from the dictionary window, the user is instantly be positioned to the first posting belonging to that field and the prefix is not shown. The user can also type a term in the dictionary box without knowing about the presence of the prefix.
The prefix will be restored just when dragging (or double clicking) on a term from the dictionary window. You should list just those fields having a prefix. *Non-listed fields will be treated as usual*.

Note that if a field is declared as having a prefix, the first *n* characters, corresponding to the prefix length, will be always cut.

555=(zero by default) switch the "All fields" dictionary option ON (0) and OFF (1).
Syntax: `555=[0|1]`

560=supply a description for a given FST tag number.
Syntax: `560=[tag number] [description]`

Since a 560 parameter line refers to just one tag, you must put a 560 **for each** tag you wish to describe.

For example:

560=24 Entire title
560=240 Words from title

This allows specifying a description to those FST lines referring non-existing FDT lines (e.g. 240). Of course it is also possible to force a description different from the corresponding FDT line.

Multiple language support

In order to support multiple languages you may provide a list of Tag description for each supported language. To open a new language section use the following syntax:

560=0 EN

this will open the English section, because in Winisis the language code EN corresponds to English. All following 560 tags will be English tags, until the end or the next Tag 0 (zero). For example:

560=0 EN
560=24 Words from title
560=70 All authors

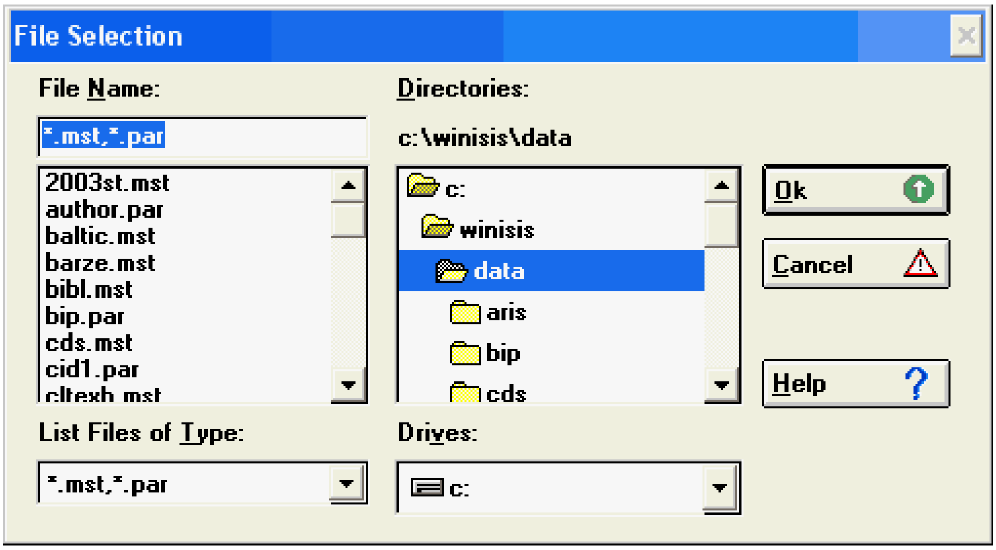
560=0 FR
560=24 Mots du titre
560=70 Les auteurs
560=69 Descripteurs

Note that the English version has no line for Tag 69. Winisis will display its corresponding FDT name.

5. CDS/ISIS Dialog Boxes

A. Open dialog box

The following dialog box is displayed each time you open a data base or a file:



File Name: *.mst,*.par

Directories: c:\winisis\data

List Files of Type: *.mst,*.par

Drives: c:

Figure 35 - Open file dialog box

i. File Name

Select or type the name of the file you want to open. This box lists files with the filename extension selected in the List Files of Type box.

ii. List Files of Type

Select the type of file you want to see in the File Name list. The file type is automatically determined by CDS/ISIS depending on the particular operation you are performing, as follows:

*.mst; *.par	Open a data base;
*.iso	Import or Export operation.
*.wsv	Saved searches
*.wpr	Saved print worksheet

iii. Drives

Select the drive that contains the file you want to open. Note that network drives will only be listed if they were connected *before* starting CDS/ISIS.

iv. Directories

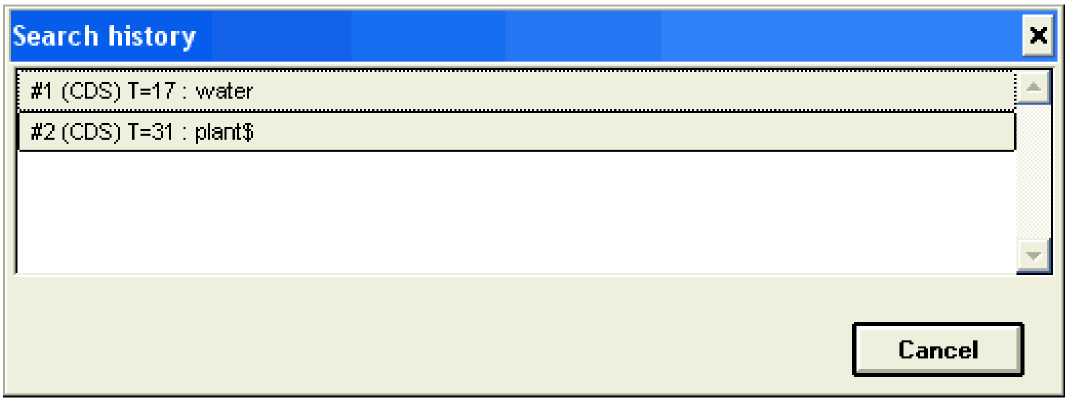
Select the directory that contains the file you want to open.

v. Buttons

Click the **OK** button to open the selected file; the **CANCEL** button to cancel the operation or the **HELP** button for on-line help on this dialog box.

B. Search History Dialog box

The search history dialog box, a sample of which is shown below, displays the list of all searches submitted so far.



#1 (CDS) T=17 : water

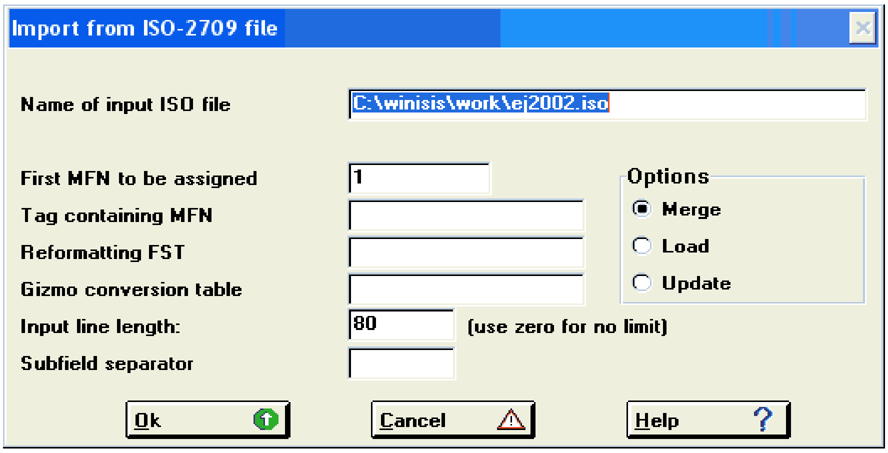
#2 (CDS) T=31 : plant$

Figure 36 - Search History Dialog box

To select a particular search click on the corresponding button. To cancel the operation click on the first button (<->).

C. Import Dialog Box

The Import Dialog box is displayed in response to the ***Import*** command of the **Data base** menu. Prior to this, CDS/ISIS displays an Open Dialog box where you select the file to be imported.



Name of input ISO file: C:\winisis\work\ej2002.iso

First MFN to be assigned: 1

Tag containing MFN:

Reformatting FST:

Gizmo conversion table:

Input line length: 80 (use zero for no limit)

Subfield separator:

Options: ● Merge ○ Load ○ Update

Figure 37 - Import Dialog box

Note: CDS/ISIS for Windows only supports ISO 2709 type 1. For more ISO 2709 types, it is recommended to use the Isis tool **ImpExp2709** available on UNESCO's website.

i. Name of input ISO file

This field contains the name of the file you are importing. The file must be in the standard ISO 2709 format as described in the *CDS/ISIS Reference Manual*.

ii. First MFN to be assigned

If you specify this parameter CDS/ISIS will renumber sequentially each input record starting from the MFN you specify. Normally you will enter 1 if you use the **Load** option, and a number which is one higher than the highest currently assigned MFN in the data base if you use the **Merge** or **Update** options.

In the latter case, however, if you specify an already assigned MFN, CDS/ISIS will automatically start numbering from the next MFN to be assigned in the data base (i.e. the **Merge** and **Update** options will behave in the same manner).

iii. Tag containing MFN

Alternatively to the option above (which will be ignored if an Input tag is specified), you may assign the MFN from a field in the input record. In this case you specify here the ISO tag of the field containing the MFN. Note that the field must contain a numeric value and may only be used for this purpose, as it will not be stored in the records of the receiving data base.

iv. Reformatting FST

This parameter is optional. If you leave it blank, the fields in the output record will retain their tag and contents. Alternatively, you may perform a certain amount of reformatting by providing the name of an FST.
Note: it is recommended to ***provide the full path and filename*** to the FST file to avoid mistakes.
The FST, when used as an import reformatting file is interpreted in the following manner:
each line of the FST represents an output field;
each output field is assigned a tag equal to the field identifier defined in the corresponding FST line;
the data extraction format given in the FST defines the contents of the field. In this format you must use the ISO tag of the fields as defined in the input file. Each line produced by the format (or each element, if the FST specifies indexing techniques 2, 3 or 4) will generate a new occurrence of the output field. Note that the stopword file of the *receiving* data base, if present, will be used in processing indexing technique 4. Assume for example that your input file contains the following fields:

```
100   Author (repeatable)
200   Title
300   Keywords (repeatable)
400   Notes
```

A reformatting FST for this file could be the following:

```
1 0 (v100/)      [output field 1 same as input field 100]
2 0 v200         [output field 2 same as input field 200]
3 0 |<|v300|>|   [output field 3 contains keywords
                  enclosed in <...>, each keyword taken
                  from one occurrence of input field 300]
```

v. Gizmo conversion table

This field may contain the name of a Gizmo conversion table to be used for import. This facility is provided to facilitate interchange of text containing diacritical marks between different hardware and/or software platforms. A gizmo conversion table is an ASCII text file, which you may create using any familiar text editor, and is defined as follows.
The first character of the first line contains the diacritical prefix (normally the @ sign for CDS/ISIS). Each subsequent line has the following format:
dxyxyxyxyxy....xy
where : **d** is the diacritical code (e.g. 7 for acute accent), **x** is the character to be accented, and **y** is the accented character.
A sample gizmo conversion table is given below:

```
    @
    6aàeè
    7aáeé
    (([
    ))]
```

CDS/ISIS uses this table to provide encoded diacriticals on export (e.g. **é** will be converted to **@7e**), and to decode encoded diacriticals on import (e.g. **@6a** will be converted to **à**). When a Gizmo conversion table is used for import, CDS/ISIS will simply strip the diacritical for all those characters not listed in the table. For example, using the table given above, the input string **@7e**. will be converted to **é**, whereas the string **@7y** will be converted to **y**, because y is not listed in line 7.

vi. Using Character conversion tables (tab files) during the import

The GIZMO line may also contain the path to a TAB file, used by Winisis to convert the input into a new character set. For example: `c:\winisis\txt_txt.tab`

vii. Input line length

This parameter specifies the length of the output/input line (CDS/ISIS default is 80). Setting the line length to zero makes it possible to read ISO file containing only a single line with no breaks.

viii. Subfield separator

Input ISO files may use a subfield separator different than the one used by CDS/ISIS. This parameter defines the separator character for subfields expected in the input file and that will be converted in the standard CDS/ISIS separator, character "^".

It is possible to specify any ASCII character as field separator, by using the following notation:

\xxx where **xxx** represents the corresponding ASCII number of the desired character. For example: for ASCII character 31, enter: **\031** in the subfield separator box.

ix. Options

This parameter indicates how the imported records are loaded into the data base:

Load: In this case the resulting data base will contain only the incoming records. Because this option erases any records which may already exist in the data base, CDS/ISIS will always request a confirmation that you actually intend to do so.

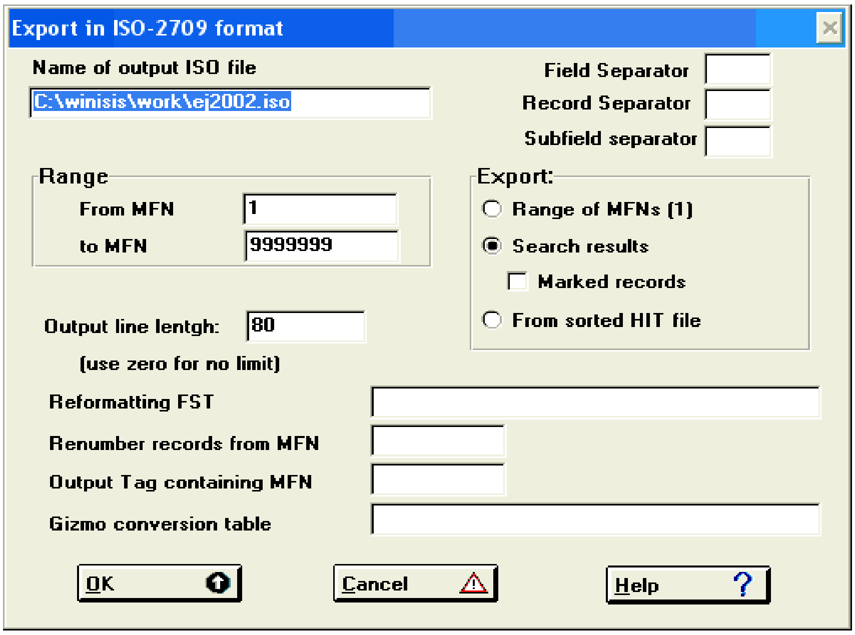
Merge: Merges the records in the input file with those already existing in the data base, if any. However, input records with the same MFN as an existing record are not replaced (unless they are marked for deletion). CDS/ISIS will display the MFN of such records, if any.

Update: The same as Merge except that records with the same MFN are automatically replaced.

Note that the ***Merge*** and ***Update*** options are functioning as described above only if you are importing the MFN of the records (i.e. you have filled the ***Tag containing MFN*** field). If not CDS/ISIS will simply add the incoming records to the Master file and assign sequential MFNs starting from the highest MFN already assigned or from the MFN you have specified in the ***First MFN to be assigned*** field.

D. Export Dialog Box

The Export Dialog box is displayed in response to the ***Export*** command of the **Data base** menu. Prior to this, CDS/ISIS displays an Open Dialog box where you select the name of the output file.



Export in ISO-2709 format

Name of output ISO file: C:\winisis\work\ej2002.iso

Field Separator

Record Separator

Subfield separator

Range

From MFN: 1

to MFN: 9999999

Export:

○ Range of MFNs (1)

● Search results

□ Marked records

○ From sorted HIT file

Output line lentgh: 80

(use zero for no limit)

Reformatting FST

Renumber records from MFN

Output Tag containing MFN

Gizmo conversion table

Figure 38 - Export Dialog box

i. Name of output ISO file

This field contains the name of the output file you have previously selected. The file will be in the standard ISO 2709 format as described in the *CDS/ISIS Reference Manual*.

ii. Field separator

This field defines the field separator character to be used in the output file. If you leave it empty, CDS/ISIS will use the standard field separator defined in ISO 2709, i.e. the ASCII character 30 (hexadecimal 1E). However, if the file to be exported is transmitted over a telecommunication line, this character may be deleted by the communication software. Therefore, CDS/ISIS allows you to redefine the field separator as a normal ASCII character which will pass through. Note that the character you select must be such that it is never used as a data character in the data base to be transmitted.

You may also specify any ASCII character as field separator, by using the following notation:

\xxx

where **xxx** represents the corresponding ASCII number of the desired character. For example: to specify as field separator the ASCII character 28, you must enter: **\028** in the field separator box.

iii. Record separator

This field defines the record separator character to be used in the output file. If you leave it empty, CDS/ISIS will use the standard field separator defined in ISO 2709, i.e. the ASCII character 29 (hexadecimal 1D). However, if the file to be exported is transmitted over a telecommunication line, this character may be deleted by the communication software. Therefore, CDS/ISIS allows you to redefine the record separator as a normal ASCII character which will pass through. Note that the character you select must be such that it is never used as a data character in the data base to be transmitted.

You may also specify any ASCII character as record separator, by using the following notation:

\xxx

where **xxx** represents the corresponding ASCII number of your desired character. For example: to specify as record separator the ASCII character 29, you must enter: **\029** in the record separator box.

iv. Subfield separator

This field defines the separator character for subfields to be used in the output file. CDS/ISIS files usually use character ^. However many bibliographic standards use character $. Default is "^".You may also specify any ASCII character as field separator, by using the following notation:

\xxx

where **xxx** represents the corresponding ASCII number of the desired character. For example: for ASCII character 31, enter: **\031** in the subfield separator box.

v. Selection parameters

You may selectively export only portions of the data base. To define the records to be exported you have two possibilities: use an MFN range or the results of the last search submitted.

To select a range of records by MFN enter the lowest and the highest MFN to be selected in the ***From MFN*** and ***to MFN*** boxes respectively. To select the last search results, first open the search set, then click on the ***Search results*** button. If you want to export marked records only click on the ***Marked records*** button (note that this option is only applicable when the ***Search results*** button is checked).

Note that if you check the ***Search results*** option and you also enter the MFN limits these will be applied to the selected records. For example, if the last search retrieved records 3, 5, 150 and 270, and the MFN limits are 10/200, only record 150 will be exported.

Another possibility is to use an existing HIT file in order to obtain a sorted ISO file. A HIT file is produced by a print run as explained in section "I. Print Dialog Box".

vi. Output line length

This parameter specifies the length of the output/input line (CDS/ISIS default is 80). Setting the line length to zero will produce a single line with no breaks.

vii. Reformatting FST

This parameter is optional. If you leave it blank, the fields in the output record will retain their tag and contents. Alternatively, you may perform a certain amount of reformatting by providing the name of an FST.

Note: it is recommended to ***provide the full path and filename*** to the FST file to avoid mistakes.

When used as an export reformatting file, the FST is interpreted in the following manner:

each line of the FST represents an output field;

each output field is assigned an ISO tag equal to the field identifier defined in the corresponding FST line;

the data extraction format given in the FST defines the contents of the field. In this format you must use the CDS/ISIS tag of the fields as defined for the data base. Each line produced by the format (or each element, if the FST specifies indexing techniques 2, 3 or 4) will generate a new occurrence of the output field.

Assume for example that your data base contains the following fields:

```
1  Author (repeatable)
2  Title
3  Keywords (repeatable)
4  Notes
```

A reformatting FST for this file could be the following:

```
1 0 mfn         [output field 1 contains the MFN]
100 0 (v1/)     [output field 100 same as input field 1(note the use of
                a repeatable group in the format to output each
                occurrence of field 1 as a separate line)]
200 0 v2        [output field 200 same as input field 2]
300 0 |<|v3|>|  [output field 300 contains keywords enclosed in <...>,
                each keyword taken from one occurrence of input field 3]
```

Note that, as none of the formats make reference to field 4, this field will *not be exported*. You may therefore use a reformatting FST to only export selected fields.

viii. Renumber records from MFN

You may reassign an MFN to output records starting from a value of your choice which you enter here. If you leave this field blank the output records retain their original MFN.

This option only affects the output ISO file. The records in the data base remain unchanged. Note, however, that this option is only effective if you are exporting the MFN by filling in the field ***Output Tag containing MFN*** below.

ix. Output Tag containing MFN

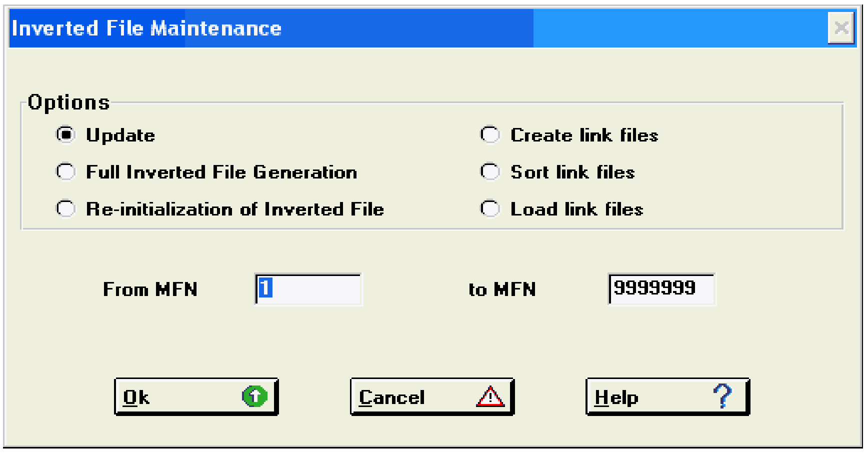
You may export the MFN as a field in the output file by assigning here an output tag. This is because the MFN does not correspond to an actual field of the data base (it has no tag and is not defined in the FDT). You may also export the MFN as a field by means of a reformatting FST, as shown in the example above.

x. Gizmo conversion table

This field may contain the name of a Gizmo conversion table to be used for export.

E. Inverted File Maintenance Dialog Box

The Inverted File Maintenance Dialog box is displayed in response to the ***I/F Update*** command of the **Data base** menu. Prior to this, CDS/ISIS displays an Open Dialog box where you select the name of the output file.



Inverted File Maintenance

Options

● Update
○ Full Inverted File Generation
○ Re-initialization of Inverted File
○ Create link files
○ Sort link files
○ Load link files

From MFN 1
to MFN 9999999

Figure 39 - Inverted File Maintenance Dialog box

i. Update

This option updates the Inverted file for all Master records for which an update is pending, i.e. records added, deleted or modified since the last Inverted file update.

ii. Full Inverted File Generation

This option generates the Inverted file of a given data base. Normally you will not use this option unless it is required. For example, whenever you want to change the contents of the Inverted file, by changing the Inverted file FST you must re-generate the Inverted file. Failing this you will introduce an inconsistency in the data base because the records added or modified after changing the FST will be inverted differently from those created before the change.

Note, however, that there are some cases which do not require a full re-inversion, even though the FST is changed. Suppose, for example, that you decide to add a new field to your data base. This requires a change in the FDT and may require a change in the FST if you also want the new field to be searchable. None of the records in the data base, however, contain yet this field and therefore a full inversion would not be required in this case, provided, of course, that you change the FST *before* modifying or adding any record containing the new field.

The Inverted file generation process consists of three steps:

1. Creation of the Link files;
2. Sorting the Link files; and
3. Loading the sorted Link files into the Inverted file.

The link files contain all the information necessary to generate the Inverted file and are created according to the Inverted file FST defined for the data base. They contain one entry for each couple search term/posting to be stored on the Inverted file. For efficiency, two link files are actually created: one containing terms up to 10 characters long and one terms longer than 10 characters. The second step sorts the Link files into alphabetical order. The purpose of this sort is to collect together all postings related to a given search term, in preparation of the last step which loads the sorted Link files into the Inverted file.
These three steps may also be executed one at a time by using the options ***Create link files***, ***Sort link files*** and ***Load link files*** in this order.

This facility is provided for large data bases where the full Inverted file generation may require a relatively long time, and considerable disk space. Note that, in this case, you must not make any change to the Master file (adding, modifying or deleting any record) until the last step has been executed.

You should be aware that CDS/ISIS does not automatically delete the Link files (.LN1, .LN2., .LK1 and .LK2).

You may delete them manually at any time, after the Inverted file generation is completed, or, if you perform the inverted file generation in three steps, you may delete the unsorted Link files (.LN1 and .LN2) after executing the ***Sort link files*** option and before executing the ***Load link files*** option.

CDS/ISIS will keep you informed of the step it is executing by displaying appropriate information messages on the screen.

iii. Re-initialization of Inverted File

This option erases the contents of the Inverted File and re-allocates all the associated files to a minimum size. You may want to use this option before re-generating the Inverted File.

iv. Create link files

This option generates the Link files as explained above.

v. Sort link files

This option sorts the Link files produced by the ***Create link files*** option. The sorted Link files may then be loaded using the ***Load link files*** option.

The process of sorting each link file (short and long terms) is the same. During the first step, called phase 0, the link file records are sorted in groups as large as can fit in memory. If the whole link file can fit in memory, then the sort is finished after phase 0. Otherwise the sorted groups are distributed on four temporary work files and then merged. Depending on the number of records, there may be one or more merge steps, called phase 1, 2 , etc., each step producing larger and larger groups of sorted records, until a single group is obtained.

vi. Load link files

This option creates the Inverted file from the sorted link files. It can be used after option ***Sort link files*** to complete the Inverted file generation.

vii. From MFN ... to MFN

You may enter in these fields a range of MFNs to be inverted. You would normally only use this feature to test your inverted file FST.

If you enter an actual range only the indicated records will be inverted, and will therefore be searchable.

F. Global Add Dialog Box

WARNING: Use this option with extreme caution. Misuse may cause irreversible damages to your data. Always make a back up of your data base, by making copies or your MST and XRF files at least.

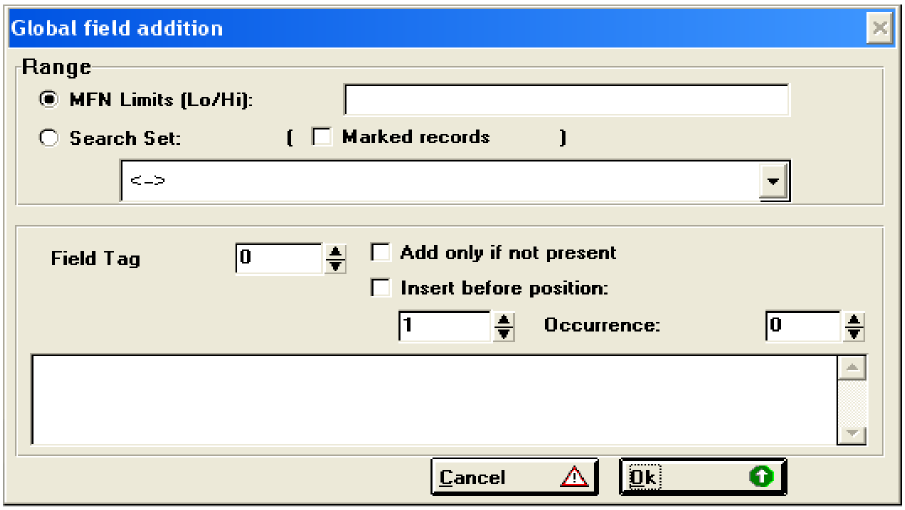


Figure 40 - Global Add dialog box

i. MFN Limits (Lo/Hi)

Enter here the lowest and highest MFN of the records to which the operation must be applied. For example: **1/20**.

ii. Search Set (marked records)

Select from the list the search set to which you would like to apply the operation. It is also possible to limit the scope to marked records (if any).

iii. Field Tag

Enter here the tag of the field to be added. Check "*add only if not present*" to limit the scope of the operation.

iv. Insert before position

Selecting this checkbox it is possible to specify where in the existing field the new text should be added. Moreover, it is possible to specify which occurrence number should be taken into account. The new parameter has the following behaviour:

- *Insert before position* checked: if the given field tag does not exist, it is automatically created with the specified content, unless the option "add only if not present" is also checked.
- *Position* specified: in case the field is long enough, the new text is inserted in the specified position. Otherwise it is appended at the end of the field.
- *Occurrence num.:* if an occurrence number is specified (greater than zero), only the specified occurrence is treated. If the field has not enough occurrences, one new occurrence is created. If "Occurrence num." is zero, all occurrences will be treated.

v. Field Contents

Enter here the contents of the field to be added.

vi. OK Button

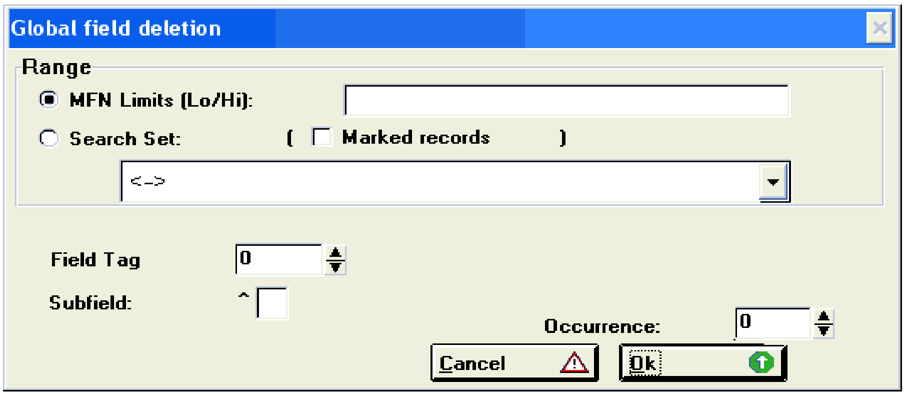
Press this button to start the operation.

vii. Cancel Button

Press this button to cancel the operation.

G. Global Delete Dialog Box

WARNING: Use this option with extreme caution. Misuse may cause irreversible damages to your data. Always make a back up of your data base, by making copies or your MST and XRF files at least.



Global field deletion

Range

● MFN Limits (Lo/Hi):

○ Search Set: (□ Marked records)

<-->

Field Tag: 0

Subfield: ^

Occurrence: 0

Figure 41 - Global Delete dialog box

i. MFN limits (from/to)

Enter here the lowest and highest MFN of the records to which the operation must be applied. For example: **1/20**.

ii. Search Set (marked records)

Select from the list the search set to which you would like to apply the operation. It is also possible to limit the scope to marked records (if any).

iii. Field Tag

Enter here the tag of the field to be deleted.

iv. Subfield

Specify a subfield identifier (one character) in order to limit the scope to that given subfield.

v. Occurrence

If greater than zero, only the specified occurrence is treated. If set to zero, all occurrences will be treated.

vi. OK Button

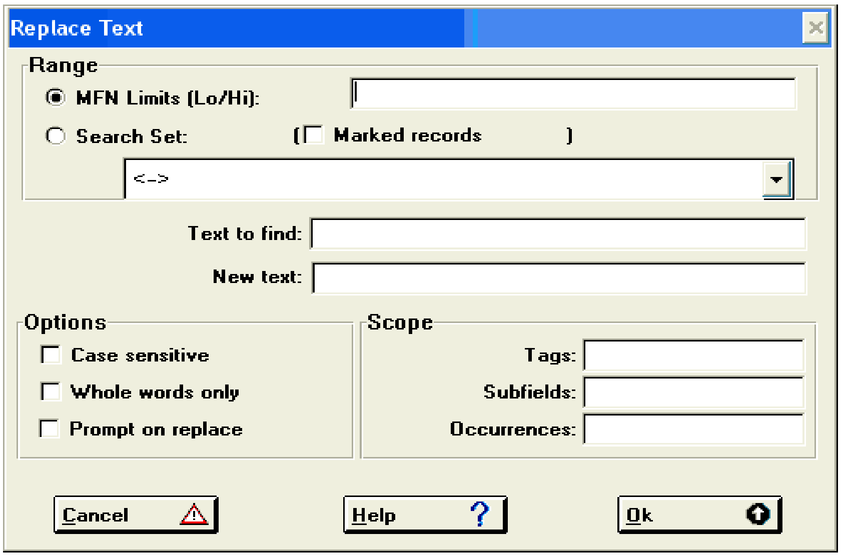
Press this button to start the operation.

vii. Cancel Button

Press this button to cancel the operation.

H. Global Replace Dialog box

WARNING: Use this option with extreme caution. Misuse may cause irreversible damages to your data. Always make a back up of your data base, by making copies or your MST and XRF files at least.



Replace Text

Range

● MFN Limits (Lo/Hi):

○ Search Set: (□ Marked records)

<->

Text to find:

New text:

Options

□ Case sensitive

□ Whole words only

□ Prompt on replace

Scope

Tags:

Subfields:

Occurrences:

Figure 42 - Global Replace Dialog box

i. Range

Specify here either the range of MFN's to be modified, by selecting ***MFN Limits*** and entering the lowest and highest MFN in the range separated by a slash, e.g. **1/100**; or a previously executed search, by selecting ***Search set*** and then selecting a specific search. In the latter case you may select only marked records by checking ***Marked records***.

ii. Text to find

Enter here the text to be replaced.

iii. New text

Enter here the replacing text. Leaving this box empty will delete the string entered in the ***Text to find*** box.

iv. Options

Check one of these options if applicable:
Case sensitive: will search the text as entered in the ***Text to find*** box. If this button is not checked, the text search will be case insensitive, e.g. 'WATER' is considered equal to 'water';
Whole words only: will only replace the text when this is preceded and followed by spaces or is at the beginning or the end of the field (or subfield).
Prompt on replace: will ask confirmation before each change.

v. Scope

If you want to replace text in certain fields only, enter the applicable tag(s), up to a maximum of 10, in the ***Tags*** box separated by comma (e.g. **100,110,120**). If you want to replace text only in certain subfields, enter the applicable subfield code(s) in the ***Subfields*** box (e.g. **abc**). It is also possible to specify to which ***occurrences*** the change should be applied: 1,2,3 etc…

I. Print Dialog Box

The Print Dialog box consists of 5 pages:

General	where you choose what and where to print and the print format you want to use;
Presentation	where you define titles, headers and footers for your document;
Margins	where you define the page printable area;
Layout	where you define the page layout of your document, such as page numbers and number of columns;
Sorting	where you define sorting criteria.

1. Guidelines for setting up a print run

Below are some basic guidelines for setting up a print run, which, at the same time, summarize the most salient features of the CDS/ISIS printing and sorting facilities (a more detailed discussion of all the features is given in the following sections).

In the ***General*** page:
Define the records to be printed: you may print the whole data base or a specific range of records, by giving the lowest and highest MFN to be printed in the MFN limits field of the print worksheet; alternatively you may print the results of a search which you have previously submitted.
Define which fields must be printed: these are specified in the format provided in the Print format field of the print worksheet. You may either use a pre-defined format or specify one directly.
Define the output medium: you may direct the output to the printer or to a disk file by setting the appropriate value in the Print file name of the print worksheet. If you intend to print the output directly on the printer, you must ensure that:

- the printer is ready and connected;
- the correct forms for your output are mounted; and
- there is enough paper available.

In the ***Presentation*** page:
Define report titles: you may provide up to three title lines which will be printed at the beginning of the output. Some or all of these may be repeated as running titles at the top of each page.

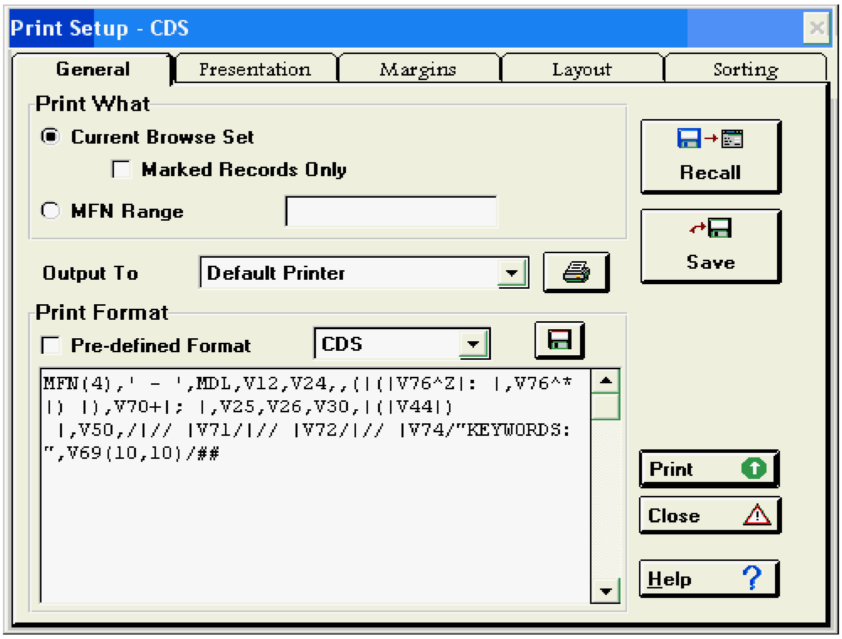
In the ***Margins*** and ***Layout*** pages:
Define the page layout: you have full control on the line width, number of lines per page and single or multi-column printing, as well as on page numbering, by setting the appropriate parameters provided in the print worksheet.

In the ***Sorting*** page:
Define the output sequence: by means of the sort worksheet you may define the exact sequence in which the records must be printed. Records may be sorted in virtually any desired combination of fields and subfields.
Define the headings: the field(s) by which the records are sorted may be used as headings. For example, if you sort the output by author, the records will be arranged in alphabetical author's name order. If you select the author to be a heading, each author's name will only be printed once for all records having the same author. The ***Sorting*** page also allows you to define whether your output will feature headings or not, and, if yes, the number of levels of headings you want. You may, for example, arrange a listing of conference papers by place (first level), date (second level) and conference title (third level).
According to the parameters you supply, CDS/ISIS will proceed as follows (steps 1-3 are bypassed if no sorting is specified):

1. using the information you provide in the ***Sorting*** page, it first builds a ***hit file*** which contains the specified sort keys and the corresponding headings;
2. it then sorts the hit file;

3. using the sorted hit file as a driver, for each record in this file it first checks whether one or more headings have changed; if so it formats and prints the required headings according to the heading format you have supplied, or the default heading format;
4. then it prints the corresponding master file record according to the print format you have defined. An exception to this is when you specify ***** instead of a print format (see below).

2. Print Dialog box (General)



Print What

● Current Browse Set

□ Marked Records Only

○ MFN Range

Output To: Default Printer

Print Format

□ Pre-defined Format: CDS

```
MFN(4),' - ',MDL,V12,V24,,(|(|V76^Z|: |,V76^*
|) |),V70+|; |,V25,V26,V30,|(|V44|)
 |,V50,/|// |V71/|// |V72/|// |V74/"KEYWORDS:
",V69(10,10)/##
```

Figure 43 - Print Dialog box (General)

i. Print What

- **Current Browse Set** Select this option if you want to print the record set you were browsing before entering the print dialog. If you were browsing a Search result (called also a Search Set), you can choose to print only the records you marked, if any, by selecting the option ***Marked records Only***.
- **MFN Range** Here you can type the range(s) of records you are interested to print. Do not enter spaces. Ranges must be in the form:

  n prints only this record
  -n prints all records form first up to this one
  n- prints all records from this one up to the maximum MFN
  n-m prints all records from MFN *n* to MFN *m*

  You can concatenate ranges using commas, for example:

  ```
  -10,15,20-30,50-
  ```

  will print the first 10 records, record 15, records 20 to 30 and all records over MFN 50 (included).

ii. Output To

After you have chosen what to print, you have to choose the output device. CDS/ISIS gives you the following options:

- **Printer** The output will be printed on the current Windows default printer.
- **Postscript File** The output will be stored in a file called *output.ps* in the Isis work directory (as specified in parameter 5 of syspar.par).
- **ASCII File (ANSI)** The output will be stored as a text file in ASCII format using the Windows character set. Note that the ampersand character (&) is not stored correctly.

- **ASCII File (OEM)** The output will be stored as a text file in ASCII format using the MS-DOS character set. Note that the ampersand character (&) is not stored correctly.

When you print to a file CDS/ISIS will ask you, before starting the print run, to provide the name of the output file by means of an **Open Dialog box**.

iii. Print Format

In this section you may choose an appropriate print format by selecting one of the pre-defined print formats. The check box ***Pre-defined format*** is to let you know if the print format shown below really corresponds to the print format name in the combo box. This is because this version of CDS/ISIS lets you modify an existing print format shortly before printing. After that, you can decide if you want to save the modified format by pushing the format save button .
If the printout you are producing is an index, you have two options:
to print one or more data elements of each record under the corresponding headings, you specify in the print format the elements to be printed; or
to only print a short reference to the records under each heading (e.g. the MFN) enter an asterisk (*) instead of a print format. In this case, instead of printing the master file record, CDS/ISIS prints the last level heading (separated by a comma from the previous occurrence and indented by the amount specified in the data indention parameter). This facility is useful for producing compact indexes where you only print a short reference to the record, such as the MFN. Note that when you use this option you must specify at least two sort keys.

iv. Recall Button

Press this button to retrieve a previously saved print worksheet.

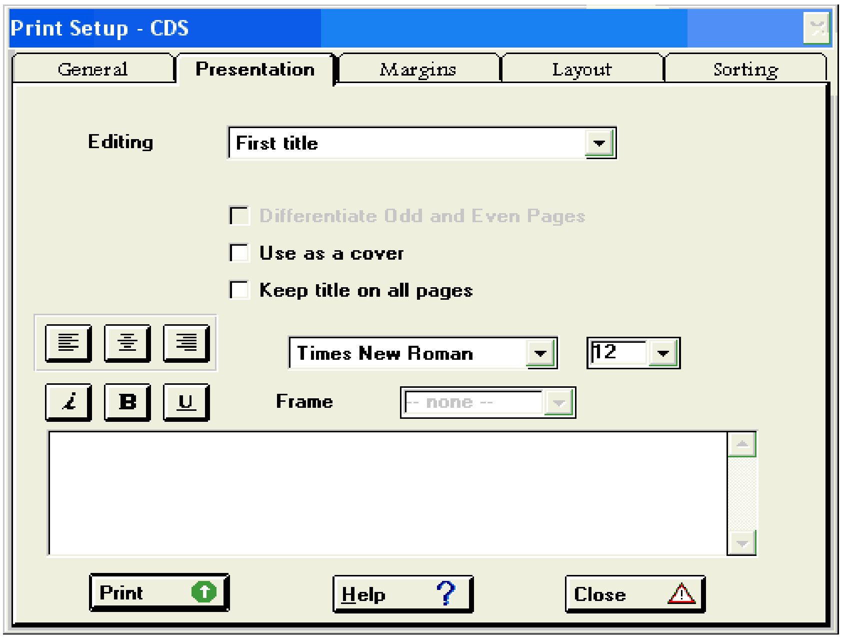
v. Save Button

Press this button to save the current print worksheet settings.

vi. Preview Button

Press this button to preview the first page of your document before it is printed.

3. Print Dialog box (Presentation)



Editing First title

□ Differentiate Odd and Even Pages
□ Use as a cover
□ Keep title on all pages

Times New Roman 12

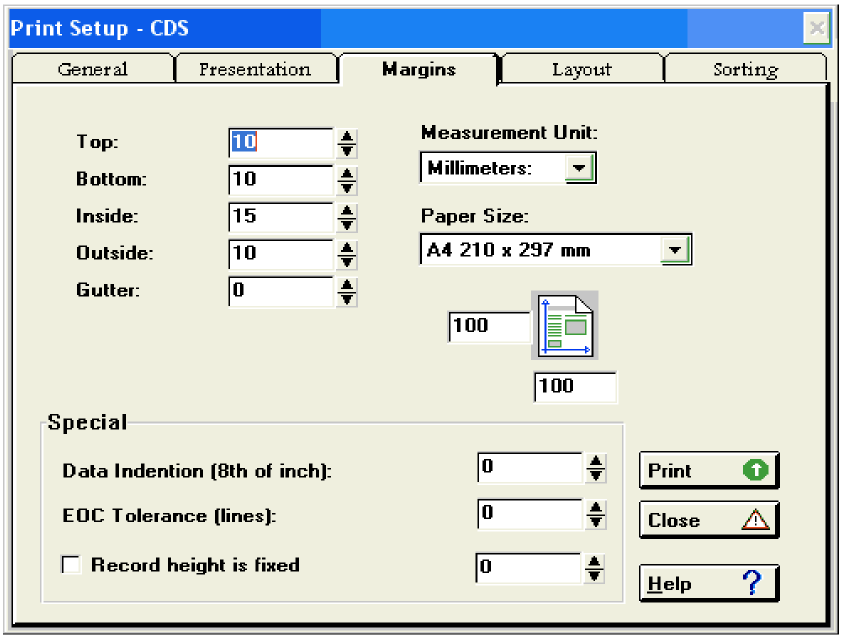
Frame -- none --

Figure 44 - Print Dialog box (Presentation)

Use this page when you want your document to have titles, headers and footers. At the top of the page a combo box shows the object currently being edited (the text is actually entered in the edit at the bottom of the window). You may define and edit any of the following items by clicking on the ***Editing*** field: **First**, **Second and Third title**, **Header**, and **Footer**. For each title you decide to add to your document, you can choose:

• to use the title as a cover page	
• to keep the title on all the pages	This is available only for the three titles.
• the title's text alignment	You can choose between: Left-aligned, Centered, Right-aligned.
• the title's font	Choose the font for the current title using this combo box.
• the title's font size	Choose the font size using this combo box or by typing in the desired size.
• the character style	You can choose your title to be Italic, Bold or Underlined or any combination of these.
• to frame the text	Not available.

4. Print Dialog box (Margins)



Top: 10
Bottom: 10
Inside: 15
Outside: 10
Gutter: 0

Measurement Unit: Millimeters:
Paper Size: A4 210 x 297 mm
100
100

Special
Data Indention (8th of inch): 0
EOC Tolerance (lines): 0
□ Record height is fixed 0

Figure 45 - Print Dialog box (Margins)

This page defines the margins to be used in your output and other related parameters.

i. Margins

The margins you can define are listed below:

Top: spacing from the top of the page to the first line printed;
Bottom: spacing from the bottom of the page to the last line printed;
Inside left margin
Outside right margin
Gutter extra space to be added to the **inside** margin of a document you plan to bind

ii. Measurement Unit

Define the measurement unit used for the various margins and the paper size. Available units are: **Millimeters, Centimeters, Inches, Twips.**

iii. Paper Size

Select the Paper Size to be used. Custom dimensions can be used.

iv. Data Indention

This parameter should normally be specified when you use headings. It indicates the indentation of each level of heading with respect to the one of a higher level and of the records with respect to the last level heading.

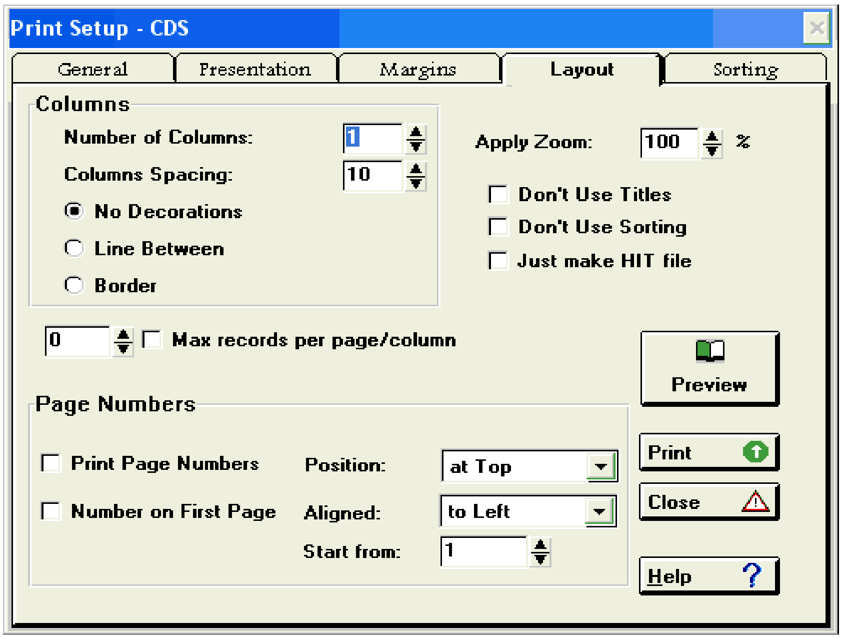
v. EOC (End of Column) tolerance

This parameter indicates the minimum number of lines which must be available on the current column (or page) before printing a record.

vi. Record height is fixed

This parameter fixes the printable area height of each record to a determined value. Useful for printing labels, etc. Note that this parameter is not precise 100%.

5. Print Dialog box (Layout)



Columns

Number of Columns: 1

Columns Spacing: 10

● No Decorations

○ Line Between

○ Border

Apply Zoom: 100 %

□ Don't Use Titles

□ Don't Use Sorting

□ Just make HIT file

0 □ Max records per page/column

Page Numbers

□ Print Page Numbers

□ Number on First Page

Position: at Top

Aligned: to Left

Start from: 1

Figure 46 - Print Dialog box (Layout)

i. Number of Columns

Number of columns per page (1 - 6).

ii. Columns Spacing

Space between columns (using the current selected measurement unit)

iii. No Decorations

No additional decorations (such as vertical lines or borders) will be printed on columns.

iv. Line Between

CDS/ISIS will draw a line between each column.

v. Border

CDS/ISIS will draw a border around each column.

vi. Print Page Numbers

Select this check box if you want CDS/ISIS to print page numbers.

vii. Number on first page

You may decide if you like numbers to be printed also on the first page. The first page is not the cover page (if you are using some cover titles)

viii. Position

Choose if you prefer numbers to appear on the top or on the bottom of the page.

ix. Alignment

Choose Left, Centered, Right, Inside, or Outside.

x. Start from

First page number to be used.

xi. Apply Zoom

This Zoom value (50-120%) will affect the print format appearance so you will be able to fit more (or less) records on a single page with no changes on your print format.

xii. Don't use titles

Tells CDS/ISIS to ignore any information in the Presentation page.

xiii. Don't use sorting

Tells CDS/ISIS to ignore the Sort page.

xiv. Just make HIT file

Tells CDS/ISIS to stop processing just after the sorting of the records. This will produce a file (so called ***HIT file***) containing the sorted keys and their corresponding MFN numbers. The HIT file can be used during the ISO 2709 Export to produce sorted ISO files. A HIT file is composed by two files:

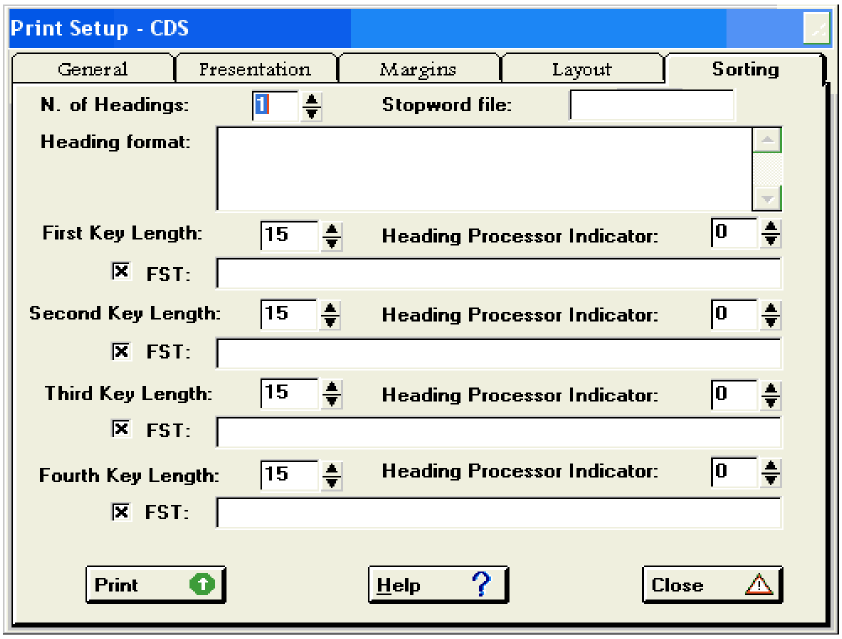
```
<dbasename>.hit
<dbasename>.hxf
```

stored in Winisis' default working directory.

xv. Max records per page/column

Print at most n record per page. Specify the highest number of record per page after which a New Page command is sent to the printer.

6. Print Dialog box (Sorting)



N. of Headings: 1

Stopword file:

Heading format:

First Key Length: 15

Heading Processor Indicator: 0

☒ FST:

Second Key Length: 15

Heading Processor Indicator: 0

☒ FST:

Third Key Length: 15

Heading Processor Indicator: 0

☒ FST:

Fourth Key Length: 15

Heading Processor Indicator: 0

☒ FST:

Figure 47 - Print Dialog box (Sorting)

This page contains all the parameters related to sorting and the handling of headings.

i. Number of headings

Enter the number of heading levels you want to use. This number must be at most equal to the number of sort keys you specify (see below). If this is not the case CDS/ISIS will set the number of levels equal to the number of sort keys you provide.

The main purpose of using headings is to help the reader to find the required information by providing a fast scanning tool: a heading *stands out* and catches the eye, thus intuitively suggesting the ordering of the listing.

As a general rule, the number of headings should be one less than the number of sort keys you specify, and the last sort key should correspond to the first data element you print for each record. For example, in a list arranged by author (first sort level) and title (second sort level) you would use one level of headings. There would not be much point in using two levels of headings, in this case, as it is unlikely that an author would write two works with the same title. As another example, you could arrange a listing of conference papers by place of conference (first sort level), date of conference (second level), title of conference (third level) and title of paper (fourth level), and use three levels of headings (place, date and conference title); the papers will then be listed in alphabetical paper title order, which you would print as the first data element in your print format.

ii. Stopword file name

If in any of the FST's used to generate the sort keys (see below) you use indexing technique 4 (word indexing), you may want to provide a stopword file, containing the list of non-significant words. In this case, enter here the name of the stopword file to be used. CDS/ISIS will ignore words listed in the stopword file while building the keys.

iii. Heading format

You may provide here your own format for printing the headings. As headings are created by means of an FST, they do not necessarily correspond to actual fields in the record (e.g. a heading may be a single *word* in a field). On the other hand, the formatting language (which is also used to format headings as well as records) has no specific command to format headings; it may only format fields or subfields. For this reason CDS/ISIS assigns to each heading a special tag which may then be used in the format to refer to the heading.

If you provide your own heading format you should note that at the time of printing, CDS/ISIS will have taken the following actions before executing it:

1. each heading is assigned a tag equal to the **field identifier** specified in the FST used to build the corresponding sort key (note however that when CDS/ISIS uses the default heading format it reassigns to each heading a sequential number starting from 1);

2. the current set of headings is then compared with the previous one and those which did not change are deleted, as this normally means that they should not be printed. You may therefore use conditional formatting to provide the required spacing. Note, however, that headings are not deleted, even if there was no change, if you specified **2** or **3** in the **Heading processing indicator** of the corresponding sort key.

Assume for example that the field identifiers assigned to the first and second sort keys are 1 and 2 respectively, the following heading format may be used to always provide a blank line before the first level heading and a blank line before the second level heading only when there is no change in the first heading:

```
MHL,""#V1(0,4)/""#N1,V2(4,8)
```

(note the use of the dummy field N1 to produce the blank line only when the first heading is missing).

Also note that when you provide your own heading format CDS/ISIS will only use the data indention parameter (supplied on the print worksheet) to offset the records printed under the last level heading. It is therefore your responsibility to provide any required indentation for the headings themselves. If you do not provide a heading format (i.e. you leave this field empty), CDS/ISIS will provide a default system format as follows:

```
MHL,""#V1(0,i)/""#V2(i,2i)/ . . . #
```

where **i** is the value you have assigned to the **data indention** parameter on the print worksheet, and V1, V2, etc. are the first, second, etc. heading. The above default format will leave one blank line before each heading and one blank line before the first record printed under the last level heading.

iv. Sort key parameters

You may specify up to 4 ***sort keys***, which will be treated in a hierarchical manner. The first, or ***primary***, sort key determines the primary sort sequence, the second sort key provides ordering for those records having the same primary sort key, etc. For example, in a listing sorted by author (primary key), you may arrange the records pertaining to a given author by title, by specifying the title as the second sort key.

You should be aware that there is an important difference between the primary key and the second and higher level keys. Whenever a master file record contains no field corresponding to the primary key, that record *will not appear in the listing*. This is normally what you would expect: for example, an author index will only include entries for those records actually having an author. However, in those cases in which you want to ensure that *all* the selected records will indeed appear in your listing, you must provide adequate alternatives for those records in which the primary key might be missing. The **IF** statement of the formatting language will help you in achieving this. Records with no field corresponding to the second or a higher level key will be sorted as blanks (i.e. they will appear at the beginning of the corresponding list) and printed normally.

For each key you must provide the three parameters described below.

Field Select Table

A Field Select Table (FST) defines the contents of the sort key and at the same time, if applicable, of the corresponding heading (remember, however, that you may define more sort keys than headings). The difference between a sort key and the heading is that the first is used to arrange records in the required order, whereas the second is used for printing. Although a sort key contains the same data as its corresponding heading, this data may be represented in a different form. For example, a sort key is always in upper case; if the field from which it was extracted contained filing information, it contains the *sortable* form of the field; etc. A heading, on the other hand, is output as specified, e.g. it will not be translated to upper case, unless you force upper case translation in the FST format. Furthermore, a heading always contains the *printable* form of a field if this contains filing information.

FSTs are discussed in detail in the *CDS/ISIS Reference Manual*. You may either supply the name of a pre-defined FST or enter one directly. If you want to use a pre-defined FST enter the name preceded by an at sign (@). The @ sign tells CDS/ISIS that this is a name, rather than an actual FST. To provide an actual FST, you must enter the three components separated by a space in the following order: field identifier, indexing technique, and format. In case you need to enter a multi-line FST, separate each line with a + sign surrounded by spaces. Here are two sample entries: the first one instructs CDS/ISIS to use the pre-defined FST called AUTHOR; the second instructs the system to create a sort key from field 10 *and* a sort key from each descriptor in field 20.

```
@AUTHOR
```

```
1 0 V10 + 1 2 V20
```

In the second example, because the FST specifies two lines, all the generated headings will be sorted in a single alphabetical sequence. In a multi-line FST you may use different field identifiers and indexing techniques for each line, e.g.:

```
1 0 V10 + 2 2 V20
```

If you specify a heading format in this case, the tag to be used in the format must take this into account. For the FST above, you must use V1 to refer to field 10 and V2 to field 20. This may be useful, for example, to print the heading using different type fonts, depending on its origin.

As mentioned above, if one or more fields selected by the FST contain filing information, CDS/ISIS will take this into account while building the sort key.

Heading processing indicator

This parameter indicates whether CDS/ISIS should build a single sort key or one sort key for each element produced by the Field Select Table. In addition, it also indicates whether headings should be unique, i.e. printed only when they change, or printed in all cases.

Enter the following:

1. **0 (zero) or 2** for single key generation: in this case the key is built from the first element produced by the Field Select Table. For example, if you sort by author you may use this option to specify that you only want the first author to be used as a sort key, and, consequently, that a given record should *only* be printed under the first author;

2. **1 or 3** for multiple key generation: in this case CDS/ISIS will build one key for each element produced by the Field Select Table. This option is normally used for the production of indexes. In this case the record from which the keys are extracted will be printed under *each* of the headings. For example you may use multiple-key generation to produce an author index or a keyword index. If you request multiple key generation in more than one level, CDS/ISIS will produce all the required combinations.

To produce unique headings, use the values **0** or **1**. The values **2** or **3** will print the headings each time, whether they have changed or not. The latter feature is particularly useful for the production of catalogue cards.

Key length

This parameter defines the size of the sort key for this print run. You should select a size which is sufficient, depending on the field from which the key is extracted, to produce the correct sequence. For example, if you specify a key length of 10 characters for a title, CDS/ISIS will only sort the first 10 characters of each title. This may have the following consequences: firstly, all titles which are different only from the eleventh position onwards will most likely be in random order as CDS/ISIS will assume that they are in fact equal; secondly, if the title is one of the headings, only the first one will be printed because CDS/ISIS prints the heading only when there is a change in the corresponding sort key.

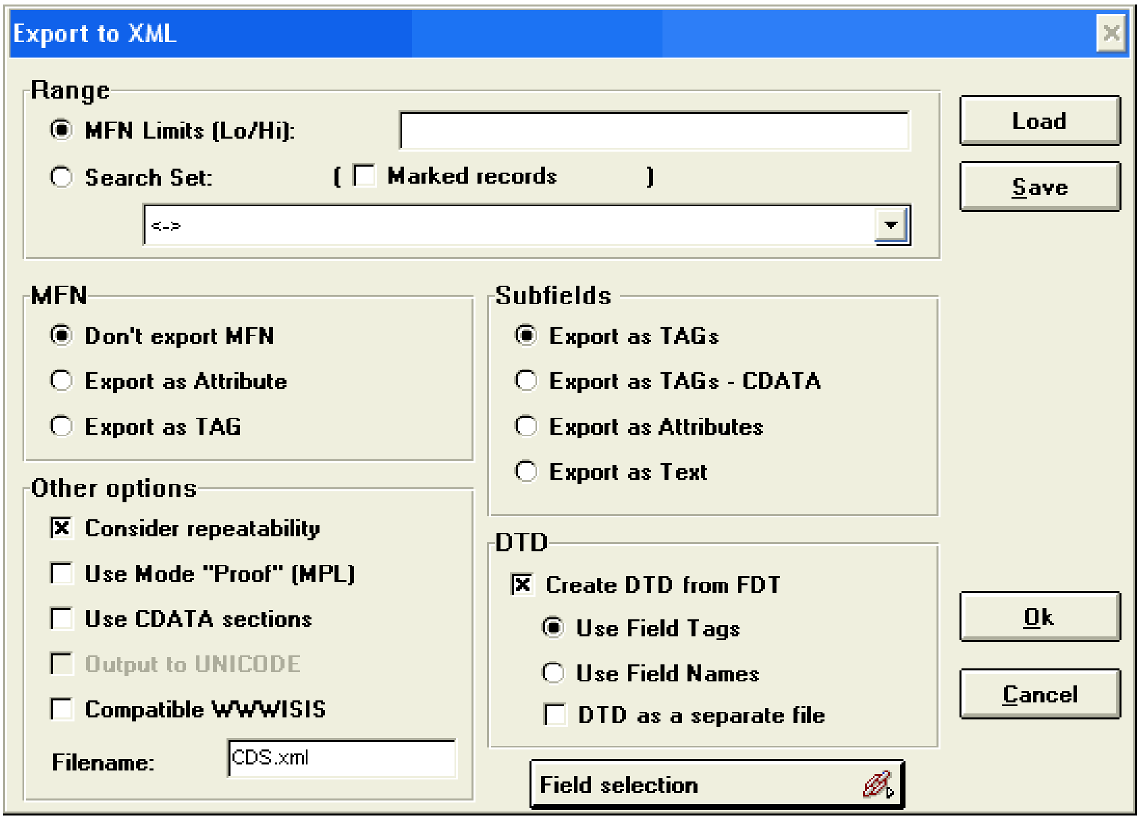
On the other hand, you should not choose a sort key length which is too large, as this will increase the file size and hence the time required for sorting. Furthermore there is a maximum limit on the total size of the sort keys, which may not exceed 4096 characters and on the hit file record size, which may not exceed 30000 characters. Thus if you need to sort on more than one key you should ensure that the total key length (for all keys) does not exceed 4096 characters and that the sum of the key length and the length of the corresponding headings does not exceed 30000 characters. If this happens, CDS/ISIS issues a warning message and ignores over-length records.

FST check boxes

This check box tells CDS/ISIS to consider (if selected) or ignore (if not) the related sort key. It may be useful when you use predefined print worksheets.

J. Export to XML

Winisis may export data in XML format using the following dialog box:



Export to XML

Range

● MFN Limits (Lo/Hi):

○ Search Set: (□ Marked records)

<->

MFN

● Don't export MFN

○ Export as Attribute

○ Export as TAG

Subfields

● Export as TAGs

○ Export as TAGs - CDATA

○ Export as Attributes

○ Export as Text

Other options

☒ Consider repeatability

□ Use Mode "Proof" (MPL)

□ Use CDATA sections

□ Output to UNICODE

□ Compatible WWWISIS

Filename: CDS.xml

DTD

☒ Create DTD from FDT

● Use Field Tags

○ Use Field Names

□ DTD as a separate file

Figure 48- Export to XML Dialog

Note: some Winisis implementations may not include the “Export to XML” option in their menus. You may access it by editing your Winisis menu and adding the following item to one of the menus:

```
Export to XML[10146]
```

i. MFN limits (from/to)

Enter here the lowest and highest MFN of the records to which the operation must be applied. For example: **1/20**.

ii. Search Set (marked records)

Select from the list the search set to which you would like to apply the operation. It is also possible to limit the scope to marked records (if any).

iii. MFN options

Here you decide if and how to export the Master File number.

iv. Subfields options

Here you decide how subfields will be represented in the final XML file and DTD.

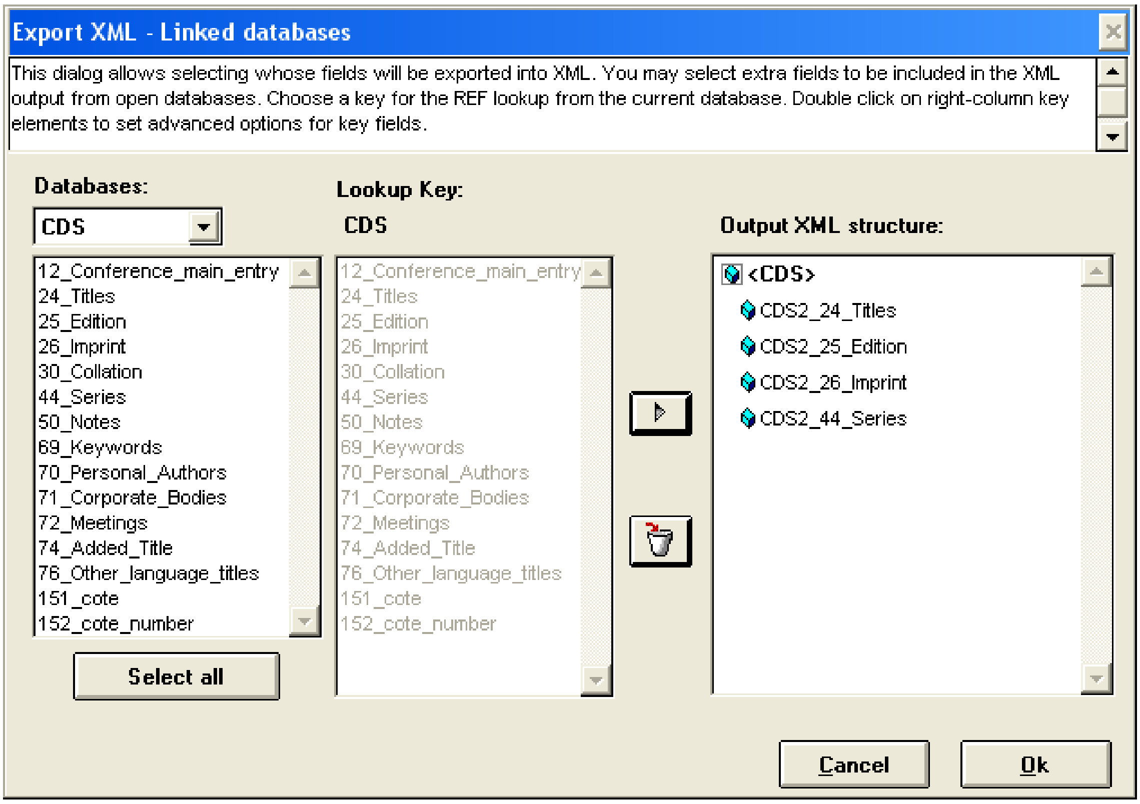
v. DTD

Select “Create DTD from FDT” if you wish to create a XML file representing your data base Field Definition Table structure.

vi. Other options

Check “Consider repeatability” if you wish repeatable fields to generate different XML tags.
Check “Use mode proof (MPL)” in order to export the information as it was entered by the user.
Check “Use CDATA sections” to avoid problems with special characters, such as < and >
Check “Compatible WWWISIS” if you wish to reuse the XML format in external web applications using WWWISIS.

Clicking “Field Selection” will open the following dialog:



Export XML - Linked databases

This dialog allows selecting whose fields will be exported into XML. You may select extra fields to be included in the XML output from open databases. Choose a key for the REF lookup from the current database. Double click on right-column key elements to set advanced options for key fields.

Databases: CDS

12_Conference_main_entry
24_Titles
25_Edition
26_Imprint
30_Collation
44_Series
50_Notes
69_Keywords
70_Personal_Authors
71_Corporate_Bodies
72_Meetings
74_Added_Title
76_Other_language_titles
151_cote
152_cote_number

Lookup Key: CDS

12_Conference_main_entry
24_Titles
25_Edition
26_Imprint
30_Collation
44_Series
50_Notes
69_Keywords
70_Personal_Authors
71_Corporate_Bodies
72_Meetings
74_Added_Title
76_Other_language_titles
151_cote
152_cote_number

Output XML structure:

<CDS>
CDS2_24_Titles
CDS2_25_Edition
CDS2_26_Imprint
CDS2_44_Series

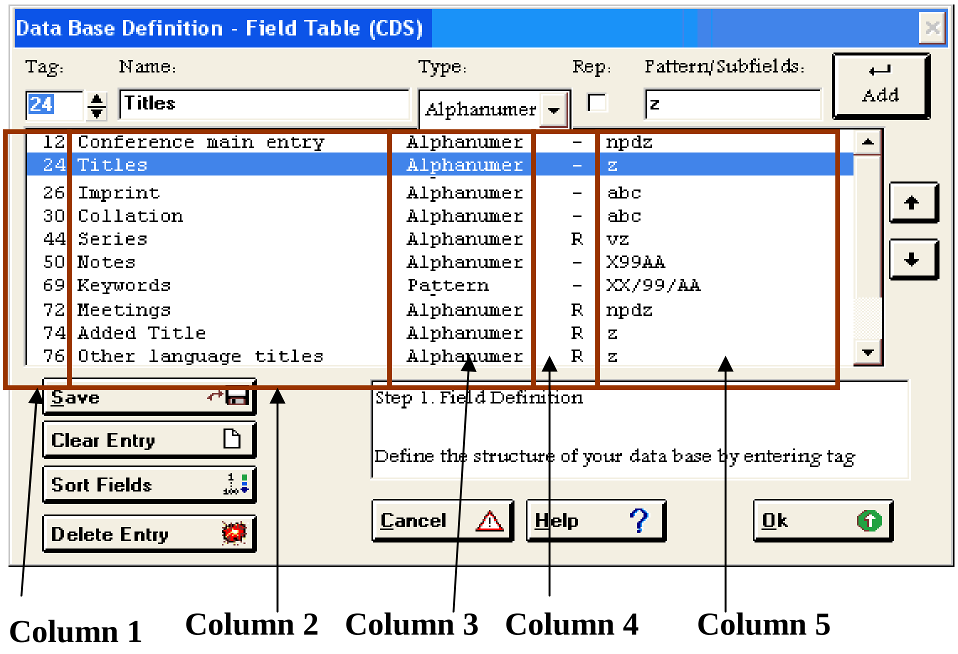
Figure 49 - XML field export options

This dialog allows you to define which fields to export (default is ALL fields) and, if possible, link two or more databases to output an unique XML file.

6. Field Definition Table (FDT)

A. Introduction

The Field Definition Table (FDT) provides information on the contents of the master records in a given data base. In particular it defines the various fields which may be present and a number of parameters for each field.
The FDT is used to control the creation of data entry worksheets for the data base and to validate the contents of fields, and it is created or modified by means of the Edit > Field Definition Table of the Main menu. A sample FDT, as displayed by the line editor, is shown below.



Data Base Definition - Field Table (CDS)

Tag: 24 Name: Titles Type: Alphanumer Rep: ☐ Pattern/Subfields: z

12	Conference main entry	Alphanumer	-	npdz
24	Titles	Alphanumer	-	z
26	Imprint	Alphanumer	-	abc
30	Collation	Alphanumer	-	abc
44	Series	Alphanumer	R	vz
50	Notes	Alphanumer	-	X99AA
69	Keywords	Pattern	-	XX/99/AA
72	Meetings	Alphanumer	R	npdz
74	Added Title	Alphanumer	R	z
76	Other language titles	Alphanumer	R	z

Step 1. Field Definition

Define the structure of your data base by entering tag

Column 1 is the tag number, column 2 is the name of the tag, column 3 is the type of the tag (numeric, alphabetic, alphanumeric) column 4 determines whether a field is repeatable or not, and column 5 indicates which subfields may be used in this tag, and if a patter is used for data entry how it should be.

B. General data base design guidelines

The generalized nature of CDS/ISIS allows you to define data bases according to your specific requirements. CDS/ISIS never makes any assumptions about the data you are processing and, in particular, it has no knowledge of their meaning. It simply provides a set of functions, normally required in any information storage and retrieval package, which help you in establishing efficient information systems. Because of this, it is impossible to provide a set of fixed rules for designing a data base, but only broad guidelines. The following paragraphs cover some basic topics on data base design. However, in order to get the most out of CDS/ISIS you should be fully familiar with all the facilities it offers, and particularly with the specific techniques described in this chapter, as a poor data base design may later prevent the use of some of the CDS/ISIS features. For example, a thorough understanding of such advanced features as the REF function of the formatting language (see under “REF(expression, format)” or the ISISPAS programming services are essential in the design of integrated data bases.

1. Data elements

A data element, as its name implies, is an *elementary* piece of information. The first step in designing your data base should be a careful and comprehensive analysis of the data elements required. Items normally eligible to be selected as data elements would be those that must be able to be processed individually. In determining this, you should ask yourself typical questions such as: "*Will the item be needed for sorting*?"; "*Must it be searchable?*"; "*Will there ever be a need to print it differently than others, e.g. in bold face or upper case?*"; etc. If the answer to any of these questions is yes, then the item should be selected as a data element.

2. Fields and subfields

Data elements may stored in fields or subfields. A field is identified by a numeric *tag* and is defined in the FDT of the data base. You may think of the tag as the *name* of the field as it is known by CDS/ISIS. Each time you want CDS/ISIS to perform an operation on a particular data element you must supply the tag of the field where that data element is stored. For example, in the FDT given above, the title is assigned tag 24. If you wanted to display the contents of the title field you would ask CDS/ISIS to display V24 (which is the formatting language command to display a field).

CDS/ISIS normally treats the contents of a field as a continuous string of characters and as a single entity. You may, however, subdivide a field into *subfields.* In this case the field contains more than one data element, each being stored in a different subfield. Unlike fields, subfields are not identified by a tag but by a *subfield delimiter.*

A subfield delimiter is a 2-character code preceding and identifying a variable length subfield within a field. It consists of the character ^ (not sign) followed by an alphabetic or numeric character, e.g., ^a.

In the FDT listed above, the Imprint field has been defined as containing the place of publication, publisher and date of publication in the three subfields a, b and c respectively. A sample Imprint could be:

```
^aParis^bUnesco^cl985
```

A field containing subfields may be accessed as a single entity, by referring only to the tag of the field (e.g. v26). In this case CDS/ISIS provides options for displaying subfield delimiters (normally for proofreading purposes), or automatically replacing them by punctuation marks. However, because subfields are identifiable through their subfield delimiter, you may also access each subfield individually, by specifying both the field tag and the relevant subfield delimiter. For example, `V26^b` refers to the Publisher subfield of the Imprint field and `V26^a` to the Place of publication subfield.

In designing a data base, remember that the CDS/ISIS formatting language has a facility for automatically replacing subfield delimiters by punctuation marks. Try, if possible, to chose delimiter codes in such a way that the replacing punctuation is suitable for the application, otherwise you will have to format each subfield individually. The standard delimiter replacement table is given under "Mode command".

Note that the first subfield of a subfielded field need not have a subfield delimiter, *provided that it is always present.* For example, if in a title field you wanted to use a subfield for the subtitle, the title part of the field, which will obviously always be present, need not have an explicit delimiter. Thus the following entry for this field would be possible:

```
Il nome della rosa^bUn manoscritto
```

3. Repeatable fields

In those cases where a given data element may occur more than once in a given record, CDS/ISIS will create as many fields as required to hold all the *occurrences* of the data element. This type of field is called a *repeatable field.* A typical example is the Author field in a bibliographic record. All the occurrences of a repeatable field have the same tag. CDS/ISIS offers facilities for handling and formatting repeatable fields. You can access a particular occurrence of a repeatable field individually through the formatting language. If, for example, it is the case, that the first occurrence of a repeatable field needs a particular treatment (e.g. the first author), it is possible.

Repeatable fields may contain subfields, which gives you a facility for handling 2-dimensional tabular data (one dimension being the field, and the other the subfields).
Furthermore, you may define a field to be repeatable even though it contains a single data element. It may be useful, for example, to be able to break down a relatively long text such as an abstract or a summary, into paragraphs to improve its legibility in a printout. By defining such a field as repeatable, you may then use the formatting language facilities, provided for repeatable fields, to indent the first line of each paragraph. Another example is when you want to be able to search such long fields by words. By entering each paragraph as a separate occurrence, you may later use the (F) operator of the search language to restrict to a paragraph the search for two or more words, which you would not be able to do if the field was not repeatable (see under "Field level and proximity search operators")

4. Control characters

Certain characters stored in a field, although keyed in as data, will be interpreted by CDS/ISIS as *control characters*, rather than data characters, and will normally activate some special type of processing. Control characters are normally reserved for CDS/ISIS use and may not therefore be used as data.
Subfield delimiters are an example of control characters. Other control characters recognised by CDS/ISIS are described below.

a. Search term delimiters

Search term delimiters may be used to identify key terms or phrases assigned to each record to enable its retrieval. The various techniques which CDS/ISIS provides to index records are described under The Field Select Table (FST). Keywords may be delimited in either of two ways: by enclosing them between a pair of slashes (/.../) or by enclosing them in triangular brackets <>. The advantage of using triangular brackets over using slashes, is that, these, unlike slashes, are reserved characters, and CDS/ISIS provides options to either display the brackets or suppress them, whereas *no option is provided to suppress slashes.* When brackets are suppressed, they are normally deleted from the displayed version of the field, except when an open bracket immediately follows a closed one: in this case CDS/ISIS will replace them with a semicolon and a space. For example, by selecting the appropriate display mode the following entry:

```
<university course> <documentation training> <library school>
```

will be displayed as follows:

```
university course; documentation training; library school.
```

Except for the case mentioned above, you must ensure that the required spaces precede and follow the open and closed bracket respectively. For example when keywords are embedded within other text in the field as below:

```
Mission report describing a <university course> in <documentation
training> at an East African <library school>
```

the spaces surrounding the keywords *must* be present in order to produce the correct display:

```
Mission report describing a university course in documentation
training at an East African library school
```

If the field was entered as follows:

```
Mission report describing a<university course>in<documentation
training>at an East African<library school>
```

CDS/ISIS would display it as:

```
Mission report describing auniversity courseindocumentation training
atan East Africanlibrary school
```

In other words, CDS/ISIS simply ignores the brackets and *does not* replace them with spaces.

b. Filing information

When producing printed catalogues you will need to sort the contents of one or more fields in order to print the records in the required sequence. CDS/ISIS will try to produce a sorting sequence according to normally accepted filing rules, but sometimes this will not be possible. In these cases CDS/ISIS offers you the possibility to state explicitly how a given field must be sorted by supplying *filing information* at the time you enter the data. Filing information is permanently recorded in the field.

This facility allows you to instruct CDS/ISIS to replace or ignore any sequence of data characters in a field whenever the field is used as a filing element, by using one of the following specifications:

<text-a = text-b>	in this case, CDS/ISIS will replace text-a by text-b when the field is used in sorting, but use text-a (and ignore text-b) when displaying the field;
<Text-a>	in this case text-a will be ignored when sorting and only used to display the field.

Below are a few cases in which this facility is normally used (but its use is not restricted to just these cases):

```
Entered as        <The> evolution of information systems
Sorted as         EVOLUTION OF INFORMATION SYSTEMS
Displayed as      The evolution of information systems

Entered as        <100=one hundred> days
Sorted as         ONE HUNDRED DAYS
Displayed as      100 days

Entered as        <Mc=mac>Pherson, J
Sorted as         MACPHERSON J
Displayed as      Mcpherson, J
```

C. FDT Parameters

Each line of the FDT defines one field of the Master file record and contains 6 parameters: the *field tag, name, length, type, repeatability* and*subfields delimiters* or *pattern* These are described below.

1. Field Tag

The tag is a unique numeric value (in the range 1-32767) identifying the field. You will use the tag of the field each time you want CDS/ISIS to perform a given operation on the field. The tag is stored in the master record and is associated with the contents of the corresponding field.

2. Field Name

The field name is a descriptive name you assign to the field. It is normally used in data entry worksheets to label the field on the screen. You may consider that this is the name of the field as you know it, whereas the tag is the name by which the field is known to CDS/ISIS. The field name can be up to 30 characters long.

While creating a data entry worksheet, CDS/ISIS will supply this name as the default name of the field.

3. Field length

This is a number (in the range 1-32000) indicating the expected length of the field. The maximum record length is 32000 characters.

4. Field type

The field type indicates possible restrictions on the data characters which may be stored in the field. The field type may be one of the following:

Alphanumeric the field may contain any alphanumeric character. X-type fields are not checked by CDS/ISIS. The definitions of alphabetic characters may be customized by each user through the system table ISISAC.TAB described under "Alphabetic characters table (ISISAC.TAB)".

Alphabetic the field may only contain alphabetic characters (note that the space is not an alphabetic character!). During data entry, CDS/ISIS will check this restriction, issue a message if it is not satisfied and request you to correct the field.

Numeric the field may only contain numeric characters (0-9). During data entry, CDS/ISIS will check this condition, issue an error message if it is not satisfied and request you to correct the field.

Pattern the field has a fixed structure controlled by an entry pattern that you define in the data entry worksheet. The entry pattern defines the type of character that can be entered in each position of the field. Patterns are discussed below.

The line editor default type is Alphanumeric.

5. Repeatability

This parameter defines whether the field is repeatable (i.e. it may occur more than once in any given record) or not. An R indicates that the field is repeatable (see, for example, fields 44, 70, 71, 72, 74 and 76 in the sample CDS/ISIS database called CDS.MST). Note that P-type (pattern) fields may not be repeatable.

The number of occurrences of a repeatable field is only limited by the maximum record length.

6. Subfields/Pattern

Depending on the type of field defined, this entry defines either the set of subfields, if any, allowed in the field (for types X, A or N), or the pattern (for type P).

a. Subfields

If the field contains subfields, the allowed subfield identifiers are defined here, in the order in which they must appear. Note that the not sign (^) identifying the subfield delimiter is not entered. For example, if a field may contain the subfields ^a ^b and ^c, these are defined in the FDT as abc (and not ^a ^b and ^c)

b. Pattern

A *pattern* is a character by character description of the contents of the field. For each field position you define in the pattern the type of character which it may contain as follows:

X the position may contain any alphanumeric character

A the position must contain an alphabetic character

9 the position must contain a numeric character

Other the position must contain the indicated character.

Note that X and A must be entered in upper case.

Here are some examples of patterns and corresponding entries:

Pattern	Entry	
99-999/AA	35-674/XE	(valid)
	35-j56/XE	(‘j ‘ is invalid)
XXX(AA)9	Xrr(BB)7	(valid)
	...(78)9	(« 78 » is invalid)

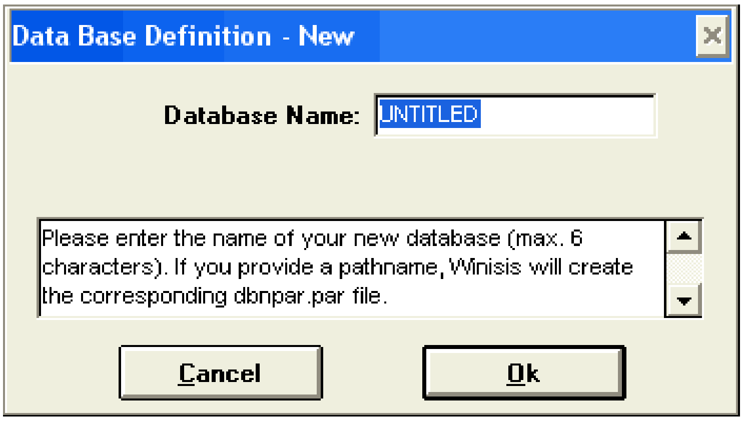
Note the following restrictions on P-type fields:

- they cannot be repeatable; and they cannot contain subfields.
- Syspar parameter 157 must be enabled

D. Creating a new database

1. The Database Definition Wizard

Using this utility, new databases can be defined and created easily, even without knowledge of the CDS/ISIS formatting language. It is called by clicking the corresponding toolbar icon or selecting menu Database > New.



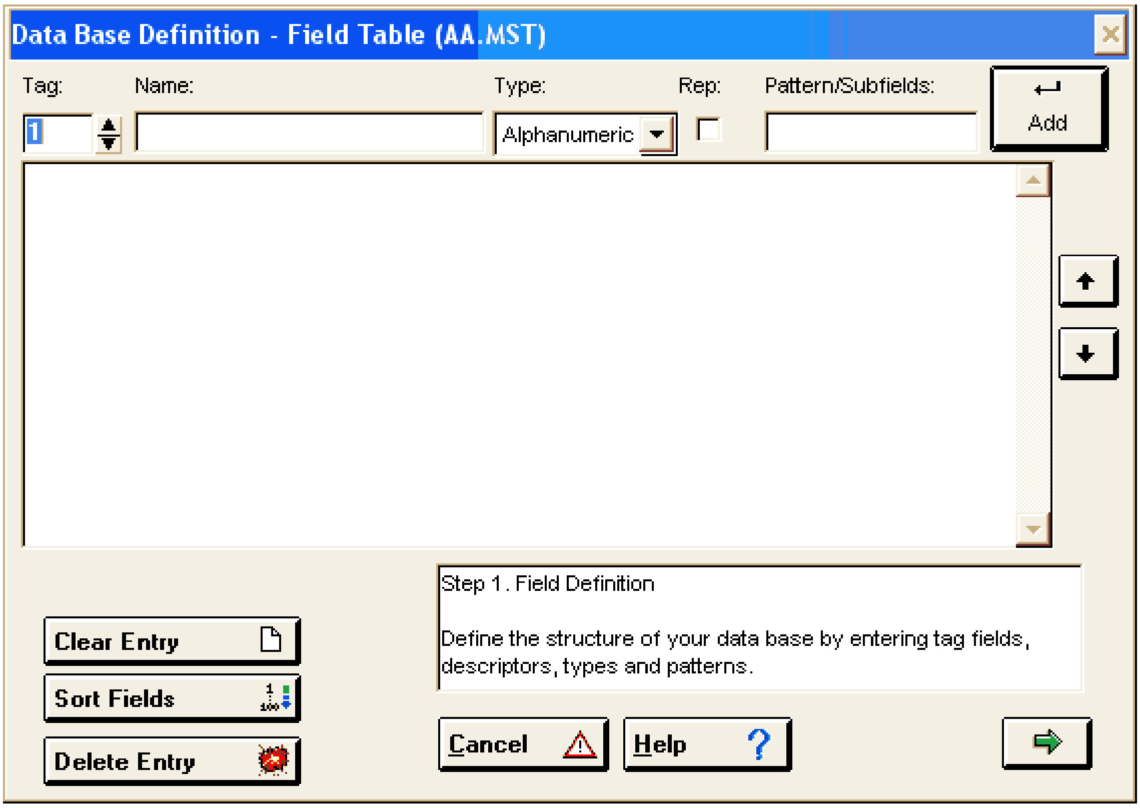
Data Base Definition - New

Database Name: UNTITLED

Please enter the name of your new database (max. 6 characters). If you provide a pathname, Winisis will create the corresponding dbnpar.par file.

Figure 50 - Database > New

You will be prompted for the name of a new database not longer than 6 characters long. In the box below the following instruction will appear. “**Please enter the name of your database (max. 6 characters)**”. If you provide a pathname, Winisis will create the corresponding dbn.PAR file. Type in the desired name of the data base. The following Window will appear:



Step 1. Field Definition

Define the structure of your data base by entering tag fields, descriptors, types and patterns.

Figure 51 - Step 1 - Field Definition Table

a. Step 1: Field definition table:

Define the structure of the new database by entering field tags, field descriptions, types, and patterns. Refer to FDT parameters for a more detailed description of the required parameters. Tag 1 is already given for the first tag number, but it can be changed to any tag number you prefer.
In the "**Name**" block type a short description of the field for the type of information to be entered there, e.g. Name of person.

In the "**Type**" block you can select the appropriate type for the field by clicking on the downward arrow. It can either be Alphanumeric, Numeric and Alphabetic. Then indicate whether the field should be repeatable, by clicking in the check mark box "**Rep**".
In the "**Pattern/Subfield**"box enter the subfield delimiters if you want to use subfields by entering the letters of numbers of the subfields (e.g. a,b,c or 1,2,3). Then click on Add, and the field will be added in the box below. Continue to add more fields and the descriptions and parameters as required.
Entries can be deleted, fields can be removed from the table, sorted, or the whole process can be cancelled. UP and DOWN arrows will change the order of the Tags. Use the "Help" button to guide you.

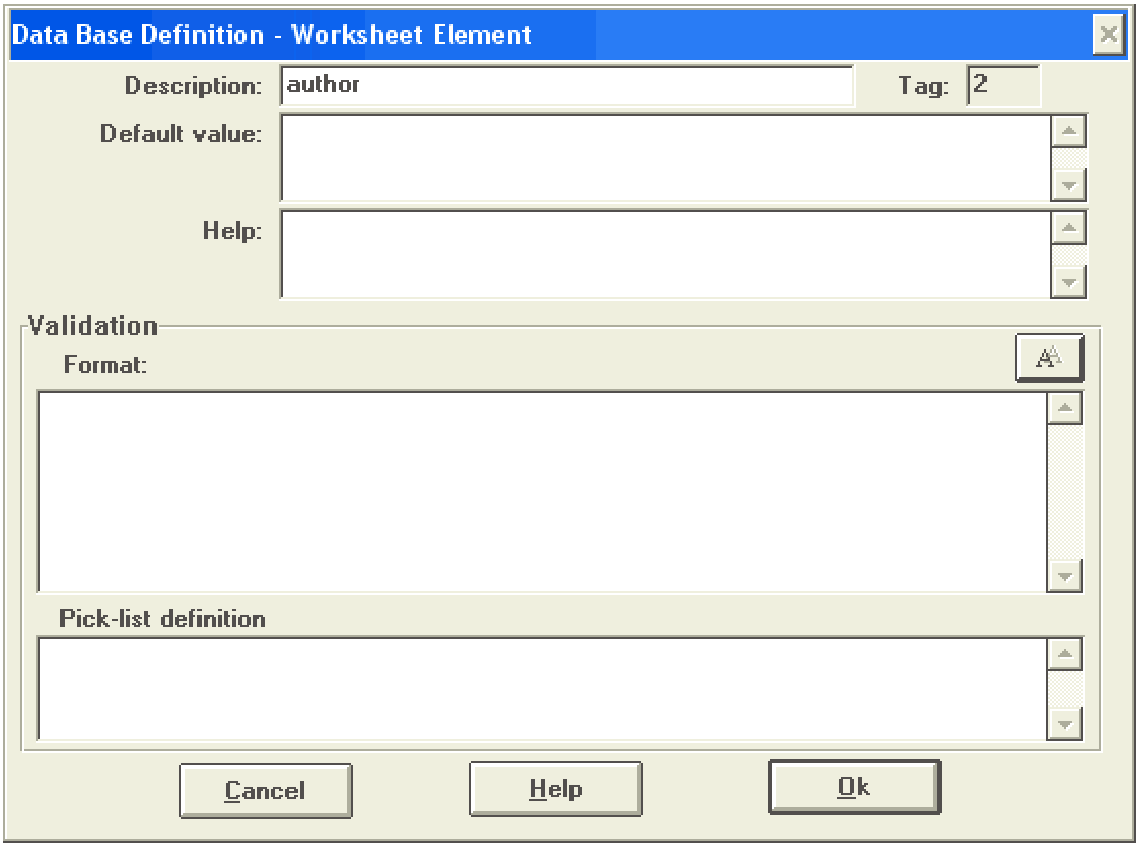
Press the right arrow ➡ to proceed to the next step.

b. Step 2: Creation of data entry worksheet

During this step the worksheet for adding/modifying records/data to the database is created. CDS/ISIS makes provision for the definition of multiple data entry worksheets for a database. By clicking on the right arrow between the two panels the field is added, and by clicking on the left arrow it is removed. The double arrow will add all defined fields to the worksheet.

It is not necessary to add all fields to a particular worksheet, or have the same order of fields as in the FDT.

By clicking the button, default values, help messages, and field validation can be defined. If all fields should be defined, check the box “**Prompt for complete data on insert**”.



Data Base Definition - Worksheet Element

Description: author

Tag: 2

Figure 52 - Data entry worksheet definition

Default values will be present when a new record is created.
The **Help message** will be displayed during data entry.
Record validation: Validation of fields will help to improve the quality of the records during data entry. Make use of the power of the CDS/ISIS formatting language, by adding commands, especially the IF - THEN - ELSE command. Click on ‘OK’ once finished with this window.
Pick-lists: it is possible to define lists of pre-defined values for this entry. Pick lists are explained in a separated document “Pick-lists.doc”.

By clicking the UP and DOWN arrows on the worksheet definition window, the order of the fields on the worksheet can be altered. Just clicking on the “**Record Validation**” button, will open a window for the definition of field validation.
The green Left arrow can be used to go back to Step 1 (Field definition), while the green Right arrow will proceed to Step 3 (Display format). The ‘**CANCEL’** button will cancel the process of defining the database.

c. Step 3: Display Format

When proceeding to this step, you will be prompted whether you want to make use of the “**Print format Assistant**”. If you answer “Yes”, this wizard will write a display format for you without any knowledge of the CDS/ISIS formatting language. All the fields of the Field definition table will be added. However, you can edit the format at a later stage by deleting unwanted fields in your display format.

If the answer is “NO”, the display format will have to be defined command by command.
Various types of display formats are then available to be Choose from:
Normal - a display format without any fancy additions

CDS/ISIS DOS Compatible Format	This format can be read by the DOS version of CDS/ISIS.
Decorated format	a display format with colours, boxes, which can only be used in CDS/ISIS for Windows.
HTML Normal	a display format in HTML
HTML Table with headers	a display format in HTML tables with table headers.

The format can be changed, e.g., the order of the fields, font, etc. By clicking the buttons on the right hand side, one can change the font properties to Italic, Bold, or Underline. The chain button will create a Hyperlink. By clicking on the right green arrow the last step will have to be done.

d. Step 4: Defining the Field Select Table (FST)

You will be prompted, whether you want to make use of the"**Dictionary Assistant**". By answering "Yes", guidance will be provided in defining the FST; if you say 'NO', you can write your own FST.
Select the fields which you want to be searchable in the database. It is not necessary to add all the fields of the FDT, but only those which you are likely to search for, e.g. Name, Town and Country. Click in the checkbox to select the searchable fields you need for information retrieval.
Then click on "OK', which will present another window, where the FST can be changed and customised. To change anything, select the line in the FST box by clicking on it, then type in any changes. If the Indexing Technique has to be changed, select the required one, by clicking on the down arrow in the select box. Nine indexing techniques are available. After the changes have been done, click on the blue "Add" button, to apply the changes in the FST line.

FST lines can be deleted with the '**Delete button**', while other tags can be added, by making use of the "**Add**" button. Some changes can be reverted with the "**Undo**" option. To go back to Step 3, use the green left button; to finish the database definition, click on "**Terminate**". Click "**Yes**" when asked "**Winisis will now create the database. Do you want to continue?** (C:\isis\data\TEST.MST)". The database will be created in this folder, and the necessary TEST.PAR will be created in c:\isis\data\, if parameter 5 in the SYSPAR.PAR file points to this folder.

Note for experienced users: it is possible and recommended to edit FST files using a external ASCII text editors, such as notepad.exe

E. Modifying the FDT

Although you may modify an FDT after it is created by using the Edit > Field Definition Table services, you should be aware that certain changes may have severe consequences, especially if you have already entered records in the corresponding data base. Additional changes, however, may be required not only to the worksheets, but also to the FST's and the display formats defined for the data base.
In case of major modifications of the FDT, changes may be required in the data base itself. If changes are required to the data base, you may be able to perform them automatically using the Global Change utilities (on the utilities menu) or the import/export services, or you may have to do them manually. See also the section on Database Definition Wizard.

Below are some hints on the impact of certain FDT changes:

Adding a field: Add the field to at least one of the data entry worksheets; add the field to at least one of the display formats; add the field to the Inverted file FST, if required.

Deleting a field: After deleting a field from the FDT, you may want to delete the field from the worksheet(s) in which it occurred by using the Edit > Data entry worksheet services, in order to prohibit further data entry. Also delete the field from the print formats and the FST in which it is used.

Repeatable field: To make a field repeatable, edit the FDT, and globally insert the separator (eg %). It may also be necessary to force CDS-ISIS to recognise the separator by exporting the data using an export FST with eg `70 0 v70+|%|` and then import again using an import FST with eg `70 0 (v70/)`

7. The Search Language

A. Introduction

The search language of CDS/ISIS is based on Boolean algebra, which provides a convenient way of expressing logical operations between classes. Each search term associated with a given record, in fact, can be viewed as representing the class of all those records associated with that term. Thus by expressing logical operations between search terms you can define precisely the class of records to be retrieved in response to your needs.

The examples given here are not meant to show how to formulate specific queries, but simply to illustrate the capabilities of the CDS/ISIS search language. However, the search language alone is not sufficient to enable you to perform a search in a specific data base. The key to efficient and effective searching is a *thorough knowledge of the search terms usable for a given data base*. It is important, therefore, that you become familiar with the ***dictionary*** of search terms available for the data base you are querying and with the rules followed to index the records in that data base.

The most efficient way to carry out a search with CDS/ISIS is through the Inverted file (the very reason to maintain an Inverted file is indeed to permit fast retrieval). You may only do this, however, if your search requirements are within the scope of the dictionary of search terms defined for a given data base, i.e. they have been anticipated when the contents of the Inverted file was defined. No matter how well thought out your data base was, however, there will still be cases where you are not able to formulate a search based on the Inverted file. For these cases, CDS/ISIS provides an alternate search method, called ***free text searching***, which allows you to satisfy *any search requirement* independently from the contents of the Inverted file. Because this second method is much less efficient it is normally used in conjunction with an Inverted file search to refine the search strategy, but may, if necessary be used independently. If you find yourself performing lengthy free text searches relatively often, it may be worth while considering whether a re-definition of the Inverted file would be called for.

B. Search expressions

1. Types of search term

In formulating a search expression you may use the three types of terms described below.

a. Precise terms

A precise term is any searchable element defined for a given data base, such as subject descriptors, key words, key phrases, words in titles, author names, etc. You must be familiar with the search terms available for each data base you intend searching.

When you use a precise search term you must specify it in exactly the same way it is known to CDS/ISIS. You should normally have a list of search terms at hand when you formulate your query (or use the **dictionary** window, to display the available search terms).

Even minor variations in spelling will cause CDS/ISIS to reject it. Thus, for example, if the term known to CDS/ISIS is COLOR (American spelling), you may not use the English spelling COLOUR as this will be rejected.

An important point to be remembered is that if a search term contains parentheses or any one of the **search operators** (*** + (G) (F) . $ ^**) or begins with a number sign (#) you must enclose it in double quotation marks (") in order to avoid a possible ambiguity. Thus if your search term is:

```
GERMANY (FEDERAL REPUBLIC)
```

you must enter it as follows:

```
"GERMANY (FEDERAL REPUBLIC)"
```

or else CDS/ISIS will issue a syntax error message.

b. Right-truncated search terms

Instead of specifying a precise search term, you may just give a ***root***. This technique, referred to as ***root searching*** or ***right truncation***, allows you to search on leading sequences of characters. CDS/ISIS will automatically perform a logical OR operation between all search terms having the specified root.

Right-truncation is indicated by placing a dollar sign (**$**) immediately after the last root character. Assume, for example, that your list of search terms contains the following set of terms:

```
FILE ORGANIZATION
FILM
FILM INDUSTRY
FILM LIBRARIES
FILM-MAKER
FILM-MAKING
FILM-MAKING TRAINING
FILMSTRIP
FILTRATION
```

Then **FILM$** is equivalent to:

```
FILM
FILM INDUSTRY
FILM LIBRARIES
FILM-MAKER
FILM-MAKING
FILM-MAKING TRAINING
FILMSTRIP
```

whereas FILM-$ is equivalent to:

```
FILM-MAKER
FILM-MAKING
FILM-MAKING TRAINING
```

As for **precise search terms**, if your root contains parentheses, any one of the **search operators**, or begins with a number sign (#) you must enclose it in double quotation marks ("). For example, **"FILM $"** is equivalent to:

```
FILM
FILM INDUSTRY
FILM LIBRARIES
```

but **FILM $** (without the surrounding double quotes) would produce an error message.

c. ANY Terms

An *ANY term* is a collective term standing for a pre-defined set of search terms. Whenever you use an ANY term in your search formulation, CDS/ISIS will automatically OR together all search terms associated with that ANY term.

An ANY term consists of the word **ANY** followed by a unique identifier, usually mnemonic, assigned to the associated set of terms. For example, the term **ANY BENELUX COUNTRY** could be used to retrieve records indexed with the individual name of any country in the Benelux group (Belgium, Netherlands, Luxembourg).

Before an ANY term can be used in a search formulation, its meaning must be defined by specifying the associated set of terms. Note that not all data bases need to implement this facility. Therefore before attempting to use an ANY term you must ensure that it is indeed available for the data base you are querying and, if so, that the ANY term you intend to use is actually defined.

2. Search operators

You may combine two or more search terms in a search expression by using ***search operators*** which indicate the intended relationship between the terms.

a. Logical OR (inclusive)

The logical OR is the class union operator. The result of a logical OR between two classes is the class obtained by merging the two classes, retaining common elements, if any, only once. Thus if A and B are two terms, representing the two classes of documents indexed with terms A and B respectively, the logical OR between these two classes is the class of documents indexed with term A or term B or both, as indicated schematically below:

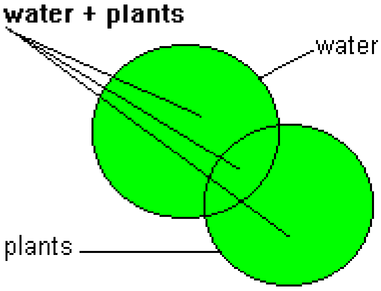


Figure 53 - Logical OR

The logical OR, therefore, is used to broaden the scope of the search and will, in general, increase the number of hits. The symbol used to indicate the logical OR operation is the plus sign (+).
Thus, for example, to retrieve documents about the Benelux countries one could use the logical OR operator as follows:

```
BELGIUM + NETHERLANDS + LUXEMBOURG
```

Note that the order in which the three countries are given is irrelevant.

b. Logical AND

The logical AND is the class intersection operator. The symbol used to indicate the logical AND operation is the asterisk (*****). The result of a logical AND between two classes is the class containing only those elements which are common to both classes. Thus, if A and B are two terms, representing the two classes of documents indexed with terms A and B respectively, the logical AND between these two classes is the class of documents indexed simultaneously with both term A and term B, as indicated schematically below:

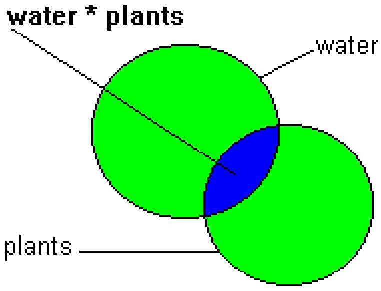


Figure 54 - Logical AND

The logical AND, therefore, is used to narrow the scope of the search by requiring the *co-occurrence* of terms and will, in general, decrease the number of hits.
Thus, for example, to retrieve documents about 'on-line information retrieval systems' you might use the logical AND operator as follows:

```
ON-LINE SYSTEMS * INFORMATION RETRIEVAL
```

Note that, as for the logical OR, the order in which the terms are given is irrelevant.

The result of a logical AND may be an ***empty*** class, i.e. a class containing no elements. In this case ***no*** records are retrieved. In the example above, the empty class would be obtained if no record in the data base contained *both* the term ON-LINE SYSTEMS *and* the term INFORMATION RETRIEVAL.

c. Field level and proximity search operators

These operators are more restrictive types of the **logical AND** operator and are particularly useful for natural language searching.

The field level and proximity search operators are:

(G)	same field (all occurrences of a repeatable field are treated as a single entity), for example:
water (G) soil	will retrieve all records with both **water** and **soil** provided that they occur in the *same* field;
(F)	same field or individual occurrence of a repeatable field, for example:
water (F) soil	will retrieve all records with both **water** and **soil** in the *same* field or the *same* occurrence of a repeatable field (**(G)** and **(F)** are equivalent when applied to non repeatable fields);
.	(period) as **(F)**, but with the additional restriction that the terms are no more than n words apart, where n is the number of periods plus one. For example:
water . soil	adjacent
water . . soil	at most one word between water and soil
water . . . soil	at most two words between water and soil
$	as **(F)**, but with the additional restriction that the terms must be *exactly* n words apart, where n is the number of dollars plus one. For example:
water $ soil	adjacent
water $ $ soil	exactly one word between water and soil
water $ $ $ soil	exactly two words between water and soil

Note that the operators **.** and **$** must be preceded and followed by one space.

d. Logical NOT

The logical NOT is the class exclusion operator. The result of a logical NOT between two classes is the class containing all the elements of the first class which do not also belong to the second class. Thus, if A and B are two terms, representing the two classes of documents indexed with terms A and B respectively, the logical NOT between A and B is the class of documents indexed with term A but not simultaneously with term B, as indicated schematically below: The symbol used to indicate the logical NOT operation is the not sign (^).

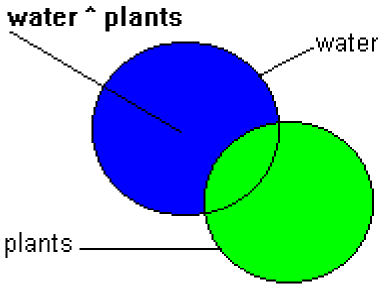


Figure 55 - Logical NOT

The logical NOT should be used with extreme caution, since it may easily result in the inadvertent loss of relevant material.

For example, a search request might concern documents about disadvantaged groups but excluding references to disadvantaged children. One might be tempted to formulate the query as follows:

DISADVANTAGED GROUP ^ DISADVANTAGED CHILDREN

It is quite possible, however, for a document on the subject to contain a section on disadvantaged children. This document would then be missed if one formulated the query as above.

Note further, that, unlike the logical OR and the logical AND, the logical NOT operation does not give the same results if the two operands are inverted. Thus, A ^ B is not the same as B ^ A (except in the particular case in which A and B represent the same class, in which case the result is the ***empty*** class).

3. Syntax of Search expressions

You may form complex search expressions by combining two or more search terms with the search operators described above.

As in normal algebra, you may use parentheses to alter the order of evaluation. In the evaluation of expressions the priority of the operators is as follows:

```
(highest)              $  and  .
    |                     (F)
    |                     (G)
    V                  *  and  ^
(lowest)                   +
```

If two or more operators of the same priority appear in the same expression within the same level of parentheses, they are executed from left to right. Thus, for example, to evaluate the following expression:

A + B * C

CDS/ISIS will first evaluate B * C and then evaluate the logical OR between A and (B * C). Whereas to evaluate the following:

(A + B) * C

it will first evaluate A + B and then the logical AND between (A + B) and C. You may *nest* parentheses, if required, as in the following example:

((A + B) * C + (D + E) + F) ^ G

In forming search expressions you must observe certain simple syntactical rules:

- no two logical operators may be adjacent to each other, except for repeats of " **.** " and " **$** " (which may not, however, be mixed together);
- the parentheses used must be balanced, i.e. the number of open parentheses must be equal to the number of closed ones and each open parenthesis must have a matching closed one.

4. Operand qualifier

You may use a ***qualifier*** to specify the field or group of fields in which you want a term to appear. This is particularly useful for data bases which may contain the same data in a number of different fields.

The qualifier has the following general formats:

```
       search term/(t1,t2,t3, . . . )
```

where **t1, t2, t3**, . . . is the set of ***field identifiers*** where you want the term to be searched.

Operand qualifiers can be used in conjunction with search operators to restrict the search to specified field(s) and can also be applied to a right-truncated search term or an ANY term. For example, consider a bibliographic data base where terms are generated, on a word by word basis, from all the fields. By simply ANDing it is possible to retrieve any records that contains the terms INTERNATIONAL, EXPERT and MEETING:

INTERNATIONAL * EXPERT * MEETING

However, the number of records that satisfy this query may be very large, as each word could appear in any of a number of fields, e.g. title, corporate body, subject, etc. By using the **(F)** operator:

```
INTERNATIONAL (F) EXPERT (F) MEETING
```

the number of records retrieved would be reduced, but you may still not be certain of the result, as the terms might appear in the summary, the title, the publisher, etc.

However, by adding the qualifier **/(62)**:

```
INTERNATIONAL (F) EXPERT (F) MEETING/(62)
```

only those records having all the terms in the same occurrence of field 62 will be retrieved (note that, in this case, it is sufficient to qualify only one of the terms). When the operand is a right-truncated search term or an ANY term, CDS/ISIS will apply the qualifier to all members of the corresponding set. Thus for example:

```
ANY BENELUX COUNTRY/(64)
```

is equivalent to:

```
BELGIUM/(64) + NETHERLANDS/(64) + LUXEMBOURG/(64)
```

5. Developing a search strategy

The basic building block of a search formulation is a **search expression**. In response to a search expression CDS/ISIS will:

- assign a unique number, called the **set number**, to the search expression you have just entered (provided it contains no syntactical errors);
- display the number of postings for each term in the expression and the number of records retrieved.

If the search expression contains an **ANY term**, the postings of all search terms assigned to the ANY term as well as the aggregate posting value for the ANY term will be displayed. Similarly, if the search expression contains a **right-truncated search term**, all the postings of the individual search terms as well as the aggregate posting value for the truncated term will be displayed.

If you use a term which is not a valid search term, CDS/ISIS will set its posting value to zero and flag the term with the message:

```
** NOT FOUND **
```

As was mentioned above, CDS/ISIS assigns a unique number to each search expression. It also saves the records matching each search expression. In later search expressions you may refer to previously entered ones by simply using the set number assigned by CDS/ISIS preceded by a number sign (#).

This facility allows you to develop your search strategy step by step. Breaking the query into elements also allows you, not only to check at each step the number of records retrieved, but also to display the retrieved records themselves. Therefore you may verify at any time the logical validity of the search formulation in terms of the relevance of the records retrieved.

Suppose, for example, that after displaying the records retrieved by the search expression (ITALY + FRANCE) * ART you realized that some irrelevant records were retrieved because they dealt with the conservation of monuments. You could then modify the formulation by introducing the following search expression:

```
#1 ^ CONSERVATION OF MONUMENTS
```

A reference to a previously defined search expression is called a ***backward reference***. Note that it is possible to apply a qualifier to a backward reference. For example,

```
#1/(64)
```

would select, from the records retrieved by search expression 1, only those in which the search expression was verified on field 64.

There are no fixed and established rules for formulating a search. You will develop your own habits as you gain experience in using CDS/ISIS. In general, an inexperienced person should not try to use complex search expressions immediately, but rather start with simple ones, which are then combined to produce the final search formulation. By referring again to the example above, the same search could have been formulated in 4 steps, as follows:

```
set number #1:          ART
set number #2:          ITALY + FRANCE
set number #3:          CONSERVATION OF MONUMENTS
set number #4:          #1 * #2 ^ #3
```

C. Free text searching

This technique allows you to specify search requirements on fields which have not been inverted and/or to specify conditions which you would not otherwise be able to indicate through the type of search expressions described above, such as the comparison of fields or comparisons of the numerical value of fields.

To perform a free text search you must first select the ***Expert Search*** command in the **Search** menu and enter your search statement in the ***Search expression*** box of the **Expert Search** Window. In order to distinguish a free text search statement from a normal search expression, it must be preceded by a question mark, as follows:

? ***Boolean expression*** or

? #n ***Boolean expression*** or

? **startMFN,endMFN Boolean expression*

where:

?	identifies this as a free text search
#n	optionally restricts the free text search to the results of a previously submitted search (**n** is the set number of the search, which may be either an Inverted file search expression, another free text search or a combination of both); if omitted the free text search is carried out over the whole data base;
Boolean expression	is a CDS/ISIS Boolean expression as defined under the paragraph "**Boolean expressions**".
startMFN,endMFN	define the scope of the free text search.

In response to a free text search, CDS/ISIS will compute the Boolean expression for each Master file record and build a hit list containing the records satisfying the specified conditions (i.e. all records for which the Boolean expression yields the value **True**). For example:

```
?  v24 : 'unesco' and val(v26^c) >= 1986
```

would retrieve all records containing **unesco** in field 24 with a date of publication (v26^c) greater or equal to 1986. The following:

```
?  #2 (p(v24) or p(v29)) and v26^b : 'unesco'
```

would retrieve, amongst the records retrieved by search expression #2, only those containing **unesco** in subfield b of field 26, provided that they contain either field 24 or field 29 or both.

Like Inverted file search expressions, each free text search is assigned a set number, which you may later use in other search expressions. Note that, although Inverted file search expressions and free text search expressions may not be mixed in the same statement, it is possible, using a backward reference, to combine any number of them in a separate search expression, as in the example below:

```
set 1      WATER * SOIL
set 2      ? #1 val(v26^c)>1985
set 3      ANY LATIN AMERICA
set 4      #2 * #3
```

Because CDS/ISIS must read each Master file record to check whether the conditions are satisfied, a free text search may be time consuming, especially when applied to a whole data base.

Another example:

```
?                           *10,5670                           v0:'Petrarca'
```

will search through records between 10 and 5670 for those containing the phrase "Petrarca". Remember that v0 (v-zero) stands for the "whole record content".

It is **not** possible to use * in conjunction with # (search result).

8. The Formatting Language

The formatting language allows you to define precise formatting requirements for data base records. Through this language you may select one or more specific data elements in the order you want and optionally insert constant text of your choice, e.g. to label some or all the fields, as well as specify vertical and horizontal spacing requirements. A collection of formatting commands in the language described in this chapter is called a **format**. In general a format defines a subset of your data base record, which may then be used by CDS/ISIS to perform a given function. Although formats are primarily used to specify the way records are displayed on the screen or the printer, they are also widely used throughout the system any time you need to perform specific operations on one or more data elements. For example, in a Field Select Table (FST), you use a format to specify to which data a given indexing technique is applied. The formatting language is therefore the core of many CDS/ISIS operations and an efficient use of CDS/ISIS requires a thorough knowledge of this technique.

To a novice, some formats may appear to be very complex, suggesting that the formatting language itself is complex. In fact, all formats, even the most complicated ones, are made up of one or more simple **commands** or **statements**, separated by commas or spaces. The apparent complexity stems from the fact that there may be many of these commands in a format. Thus the key to understanding formats is to analyse each command individually.
Although all formats are defined using the same formatting language, they may be categorized, depending on their intended use, as follows:

Display formats: used for displaying records on the screen or printing records on the printer (in the latter case they are referred to as print formats);

Extraction formats: used in FSTs to define the data to be indexed.

When CDS/ISIS processes a format, it works with three objects: a data base **record**, the ***format and a* work area** where the output produced by the format is stored. The commands are executed sequentially in the order they are listed in the format. Some commands produce data (e.g. the contents of a given field), while others produce actions (such as skipping to a new line, leaving one or more blank lines, etc.). The data produced are stored as lines of text in the work area, which is then passed to the relevant program for further processing, e.g. for printing.

When a format is used to display data, the lines produced are normally restricted to a certain maximum length (the ***line width),*** which is determined by the size of the current window.
Unless new lines are forced by explicit formatting commands, CDS/ISIS will output data sequentially and try to fill each line as much as possible. If a given field exceeds the line width, CDS/ISIS will split it into as many lines as necessary. When CDS/ISIS breaks data into lines, the break always occurs at word level, i.e. a word will never be split between two lines. All formatting commands may be entered in upper or lower case or a combination of the two.

Unless otherwise noted, all the formatting examples in the following sections refer to the sample record given below, where the content of each field is given as actually stored in the record. This record is taken from the sample CDS data base contained in the original CDS/ISIS installation as supplied by UNESCO.
`MFN = 4`

Tag	**Contents**
24	<An> Electric hygrometer apparatus for measuring water-vapour loss from plants in the field
26	^aParis^bUnesco^cl965
30	^ap. 247-257^billus.
44	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

50	Incl. Bibl.
69	Paper On: <hygrometers><plant transpiration><moisture><water balance>
70	Grieve, B.J.
70	Went, F.W.

A. Field Selectors

Field selectors are commands used to extract a specific field or subfield from a record. A special command allows you to extract the MFN of the record, even though the MFN is not, properly speaking, a field (the MFN has no tag and is not defined in the FDT).

1. Field command

To extract a field from a record code the letter **V** followed by the **tag** of the field to be extracted. The V (a mnemonic code for Variable length field) is the command telling CDS/ISIS that you want to extract a field. It may be entered indifferently in upper or lower case. Below are some examples.

Format	**Output**
v24	<An> Electric hygrometer apparatus for measuring water-vapour loss from plants in the field
v26	^aParis^bUnesco^cl965
v30	^ap. 247-257^billus.
v44	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

2. Subfield command

To extract a particular subfield from a given field just append the corresponding subfield delimiter to the tag, as shown in below. Note that you may use the special subfield delimiter ^* to select the ***first*** subfield, whichever this may be. In this case the first subfield need not be preceded by an actual subfield delimiter. Note that, as alphabetic subfield delimiters are case insensitive, you may enter the subfield delimiter code in either upper or lower case.

Format	**Output**
v26^a	Paris
v26^b	Unesco
v30^a	p. 247-257
V26^*	Paris
v44^*	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

3. Extracting a fragment of a field or subfield

You may need, in some cases, to extract a portion of a field which is not a subfield, particularly when the field has a fixed format throughout a data base (e.g. a standardized date of the form YY-MM-DD). You

may do this by appending the offset/length command immediately after the field or subfield command to which it applies. This command may be coded ***offset.length** or ***offset** or **.length**, where:
***offset** indicates the position of the first character to be extracted from the field or subfield (character positions are counted from zero, i.e. the first character is at position 0, the second at position 1, etc.); if omitted, CDS/ISIS assumes the offset to be zero;
.length indicates the number of characters to be extracted; if omitted the remainder of the field (or subfield) starting from offset will be taken.
Some examples of this command are given below, where it is assumed that the sample record also contains a field 1 as follows:

```
99-Nov-05
```

Format	**Output**
`V1*3.3`	`Nov`
`V1.2`	`99`
`Vl*7`	`05`
`Vl*7,vl*2.4`	`05-Nov`
`vl*7,vl*2.5,vl.2`	`05-Nov-99`
`v26.3`	`^aP`
`v26^b*2.4`	`esco`

Note, in the last two examples, the difference in handling a subfielded field: when this is referenced as a field (e.g. v26), offset zero represents the first actual position of the field, whereas when a specific subfield is selected (e.g. v26^b), offset zero represents *the first data character* after the subfield delimiter.

4. Field occurrences

It is possible to access individual occurrences of a repeatable field by specifying the occurrence number or range, enclosed in square brackets, immediately following the field selector. For example:

`v10[1]`	retrieves the first occurrence of field 10
`v10[2..4]`	retrieves the 2nd through the 4th occurrence of field 10
`v10[3..]`	retrieves the 3rd through the last occurrence of field 10
`v10[1]^a`	retrieves subfield ^a in the 1st occurrence of field 10
`v10[2]*4.3`	retrieves a fragment of the 2nd occurrence of field 10

It is coded as follows:
[<index> [..<upper index>]]
<index> and <upper index> refer to the first (or unique) and last occurrences, respectively. If the specified <index> is greater than the actual number of occurrences, no output is generated. The same occurs if data field is not repeatable and <index> is set to a number equal or greater than 2. However, if <index> is set to 1 and it is used in a non-repeatable field, content is normally output. This component must be used outside a repeatable group; otherwise, <upper index> is ignored. If double dot (..) is used and <upper index> is missing LAST is assumed. The LAST keyword is set with the value of total ooccurrences of a data field.

Examples:

Format	Output
V70[2]	Wynter, Hector
V70[2..5]+|; |	Wynter, Hector; Faure, Edgar
V70[1..]|; |	Jóború, Magda; Wynter, Hector; Faure, Edgar;
v70[1..3]|; |	Jóború, Magda; Wynter, Hector; Faure, Edgar;
V70[3]	Faure, Edgar
"AUTHORS: "v70[2..]|; |	AUTHORS: Wynter, Hector; Faure, Edgar;

5. Indentation command

When CDS/ISIS executes a field or subfield command it will normally output the field contents at the current line position, which depends on the last command executed. If the field cannot be fully contained in the current line, CDS/ISIS will create as many additional lines as required. Normally the continuation lines begin at line position 1. You may alter this by providing an indentation command, which must immediately follow the field (or subfield) command. The indentation command is coded **(f,c)** or **(f),** where:

f indicates the number of spaces to be left from the left margin before formatting the first (or only) line of the field. It is only effective if the field is formatted *at the beginning of a line,* otherwise it is ignored;

c indicates the number of spaces to be left from the left margin before formatting all continuation lines of a field formatted on more than one line.

A value of zero may be specified for either **f** or **c**. If only **f** is needed, you may omit **c** (CDS/ISIS will supply zero by default). However, if **c** is required you must also specify **f**. Some examples are given below.

Format	Output
V44	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium
V44(10)	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium
V44(5,9)	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium
V44(0,8)	Methodology of plant eco-physiology: proceedings of the Montpellier Symposium

6. MFN command

To extract the MFN of a record code the following:

`MFN` or `MFN(d)`

where d is the number of digits to be displayed. If (d) is omitted 6 digits will be displayed by default.

Format	Output
MFN	000004
MFN(3)	004
MFN(2)	04
MFN(1)	4

Note that you may use the F function (see under `F(expr-1,expr-2,expr-3))` to suppress the leading zeros.

B. Mode command

CDS/ISIS may display data in three different modes:

proof mode: in this mode, fields are displayed exactly as they are stored in the record. Note that CDS/ISIS *does not* insert any separator between fields or occurrences of a repeatable field. It is therefore your responsibility to ensure adequate separation of fields by using spacing commands, literals or repeatable groups as appropriate (see under “Horizontal and vertical spacing commands”, “Literals”, and “Repeatable groups”). This mode is normally used to display records for proofreading purposes.

heading mode: this mode is normally used for headings when printing catalogues and indexes. All control characters embedded in the data, such as filing information (see under “Filing information”) and descriptor delimiters (< and >) are ignored (except as noted below), whereas subfield delimiters are replaced by punctuation (see below).

data mode: this mode is similar to heading mode, but, in addition, each field is automatically suffixed with a full stop (.) followed by two spaces (or just two spaces if the field already ends with a punctuation mark). The repeatable field separator is replaced with a full stop and a space (.) or just a space if the occurrence already ends with punctuation mark. Note, however, that this automatic punctuation is suppressed if the field selector is followed by a suffix-literal (see under “Literals”).

When CDS/ISIS formats a subfielded field in heading or data mode it will automatically replace embedded subfield delimiters by punctuation marks (the initial subfield delimiter, if any, is always ignored). Furthermore, the special character combination **”> <”** is replaced by **“; ”** thus providing a simple way to format fields containing lists of key phrases enclosed in triangular brackets (and saving keystrokes during data entry). The standard subfield delimiter replacement table provided is as follows:

^a	replaced by “; ”
^b through to ^i	replaced by “, ”
all others	replaced by “. ”

A **mode command** is coded ***Mmc***, where:

m specifies the mode as follows:

P	proof mode
H	heading mode
D	data mode

c specifies case translation as follows:

U	data are converted to upper case
L	data are left unchanged

A mode command may appear as many times as necessary in a format, each remaining in effect until it is changed by a subsequent one. In the absence of an explicit mode command, CDS/ISIS will use MPL by default (proof mode, no upper case conversion). Examples of mode commands are given in the table below.

Format	**Output**
`mpl,v24`	`<An> Electric hygrometer apparatus for measuring water-vapour loss from plants in the field`
`mhl,v24`	`An Electric hygrometer apparatus for measuring water-vapour loss from plants in the field`
`mdl,v24`	`An Electric hygrometer apparatus for measuring water-`

	vapour loss from plants in the field.
mdu,v24	AN ELECTRIC HYGROMETER APPARATUS FOR MEASURING WATER-VAPOUR LOSS FROM PLANTS IN THE FIELD.
mpl,v26	^aParis^bUnesco^cl965
mhl,v26	Paris, UNESCO, 1965
mdu,v26	PARIS, UNESCO, 1965.
mpl,v69	Paper on: <hygrometers><plant transpiration><moisture><water balance>
mdl,v26	Paper on: hygrometers; plant transpiration; moisture; water balance.
mdl,v70	Grieve, B.J. Went, F.W.

C. Horizontal and vertical spacing commands

The formatting language provides five commands to control horizontal and vertical spacing. They are summarized in the table below:
The **Xn command** inserts n spaces before formatting the next data. However, if less than n positions are available on the current line, CDS/ISIS will simply skip to a new line. Thus, for example, if the next available position on the current line is 77 and the defined line width is 80, the execution of the command X7 will cause the next data to be formatted at the beginning of the next line (and not at the third position of the next line). (not supported in WINISIS graphical mode, where it has same effect as TAB)

The **Cn command** causes the next data to be formatted starting from position n of the current line. If the current line position is greater than n, then the next data will be formatted starting on position n of the following line. This facility allows you to produce tabular output. Note that if n is greater than the line width, the command is ignored.

The / **command** is similar to a carriage return on a typewriter, i.e. it forces a new line and causes therefore the next data to be formatted at the beginning of a line. However, unlike a carriage return, multiple adjacent / commands, although syntactically correct, have the same effect as a single / command, i.e. a / will never produce blank lines.

The **# command** is provided for this purpose. It performs the same function as the /, but the skipping to a new line is unconditional. Thus you may use the combination / # to ensure that one (and only one) blank line will appear on the output (note that the combination ## may cause one or two blank lines to be inserted depending on whether the line being formatted when the first # is executed is empty or not).
The use of the # command may cause a problem in those cases where the fields selected may be absent. This situation is best illustrated through the following example:

```
/#V10/#V20/#V30
```

If all fields are present in the record, the result will be that fields 10, 20 and 30 will each start on a new line and be preceded by a single blank line. However, if field 20 is missing there will be two blank lines between field 10 and field 30. This may be undesirable: if, in fact, what you want is a single blank line between each field regardless of the presence or absence of some of the fields, then the above format will not produce the desired results.
The % command is provided to solve this problem. Its effect is to suppress all contiguous blank lines (if any), existing between the current line and the last non-blank line, at the time this command is executed. Thus the following format:

```
%##V10%##v20%##v30 . . . . . .
```

will produce one and only one blank line between each field even when one or more of them are absent from a given record. A line with absent fields and an unattached Cn or Xn command is not empty; dummy field selectors may be used to prevent the output of spaces (see E. Dummy field selectors).

D. Literals

A *literal* is a string of characters, enclosed between appropriate delimiters, which will be inserted *as is* in the output. Literals may be used, for example, to label fields.
Three types of literals may be specified:

conditional literals: define text which will be output only if the associated field is present in the record. If the associated field selector is a subfield command (e.g. v24^a), the text will be included only if the requested subfield is present in the field. If the associated field selector specifies a repeatable field, the text will only be included once, regardless of the number of occurrences of the field. Conditional literals are enclosed in double quotation marks ("), e.g. **`"Title: "`**

repeatable literals: like conditional literals, they define text to be output only if the associated field or subfield is present in the record. If the field is repeatable, however, the literal will be repeated for *each* occurrence of the field. Repeatable literals are enclosed in vertical bars (|) e.g. **`|Author: |`**

unconditional literals: define text which is always output regardless of the presence of fields. Unconditional literals are enclosed in single quotes ('), e.g. **`'Summary'`**. As unconditional literals are always output as a single block of text (i.e. they cannot be split between two lines), their length should not exceed the line width otherwise they will be truncated. To output text exceeding one line, you should break it down into two or more literals. You may also provide any required indentation by using the Cn command.

Note that a literal must not contain the same literal delimiter, e.g. an unconditional literal cannot contain a single quote (although it may contain a double quote and/or a vertical bar).

Conditional and/or repeatable literals are associated to a field or a subfield by their position in the format: Literals *preceding* a field selector (also called ***prefix-literals)*** will be output before the field contents, whereas literals *following* the field selector (also called ***suffix-literals)*** will be output after the field contents.

If a repeatable prefix-literal is immediately followed by a "+" sign (e.g. **`|xxx|+`**) it will be output before *all but the first* occurrence of the field.

If a repeatable suffix-literal is immediately preceded by a "+" sign (e.g. **`+|xxx|`**) it will be output after *all but the last* occurrence of the field.

Repeatable prefix-literals and all suffix-literals are formatted as if they were physically part of the associated field contents, and obey therefore the field indentation command, if any. Conditional prefix-literals do not inherit the field indentation (you may however use the Cn command to provide indentation, if required).

A given field may be associated with more than one literal. In this case the various literals must be specified following the rules and the order given below:

Prefix-Literals

1. One or more conditional prefix-Literals. A conditional prefix-literal may be followed by other conditional prefix-Literals, vertical and horizontal spacing commands, mode commands. All commands between the first conditional prefix-literal and the associated field selector become themselves conditional and will only be executed if the field is present, otherwise they will be ignored.
2. One, and only one, repeatable prefix-literal. This, if present, must immediately precede the associated field selector.

Suffix-Literals

1. One, and only one, repeatable suffix-literal. This, if present, must immediately follow the associated field selector.
2. One, and only one, conditional suffix-literal. If present, it must immediately follow the repeatable suffix-literal, if any, or the associated field selector.
3. Suffix-literals must not be separated by commas, and there must be no comma between the field selector and the first suffix-literal: a comma signifies the end of the suffix-literals associated with a given field selector.

Null literals (i.e. zero-length literals such as "" or | |) are allowed, and may be used, for example, as prefix-literals, to provide conditional vertical spacing or, as suffix-literals, to temporarily suppress the automatic final punctuation that CDS/ISIS supplies when data mode is active.

Unlike in CDS/ISIS for DOS, in Winisis literals will not honour upper case translation if set by a preceding mode command. Examples of the different types of literals are given below:

Format	**Output**

`'MFN: ',mfn(3)/`
`mdl,"Title: "v24(0,7)`
	`MFN: 004`
	`Title: An Electric hygrometer apparatus`
	`       for measuring water-vapour loss`
	`       from plants in the field.`

`'MFN: ',mfn(3)/mdl,`
`"Title: ",mdu,v24(0,7)`
	`MFN: 004`
	`Title: AN ELECTRIC HYGROMETER APPARATUS`
	`       FOR MEASURING WATER-VAPOUR LOSS`
	`       FROM PLANTS IN THE FIELD.`

`'MFN: ',mfn(3)/mdu,`
` "Title: ",v24(0,7)`
	`MFN: 004`
	`Title: AN ELECTRIC HYGROMETER APPARATUS`
	`       FOR MEASURING WATER-VAPOUR LOSS`
	`       FROM PLANTS IN THE FIELD.`

`v70`	`Grieve, B.J.Went, F.W.`
`v70|; |`	`Grieve, B.J.; Went, F.W.;`
`v70+|; |`	`Grieve, B.J.; Went, F.W.`
`|; |v70`	`; Grieve, B.J.; Went, F.W.`
`|; |+v70`	`Grieve, B.J.; Went, F.W.`

`"Authors"/v70(3,3)+|; |`	`Authors`
	`   Grieve, B.J.; Went, F.W.`

`"(by: "v70+|; |")"`	`(by: Grieve, B.J.; Went, F.W.)`
`mdl,v26`	`Paris, Unesco, 1965.`
`mdl,v26""`	`Paris, Unesco, 1965`
`mdl,v26,""/#v99,v30^a`	`Paris, Unesco, 1965. p. 247-257.`

`mdl,v26,""/#v44|: | ,v30^a`	`Paris, Unesco, 1965.`
	
	`Methodology of plant eco-physiology:`
	`proceedings of the Montpellier`
	`Symposium: p. 247-257.`

E. *Dummy field selectors*

A dummy field selector allows the conditional output of a literal based on the presence or absence of a given field or subfield without printing the contents of the associated field. Dummy field selectors are coded as follows:

Dt or **Dt^x** or **Nt** or **Nt^x** where:

D or **N** identifies this as a dummy field selector. D indicates that all associated conditional literals must be printed only if the field is present. N indicates that they must be output only if the field is absent;

t is the tag of the field controlling the output of literals;

^x is an optional subfield delimiter code. When given, it indicates that the output of literals is controlled by the presence or absence of the specified subfield (note, however, that the absence of a field also implies the absence of a specific subfield of that field). A dummy field selector is normally preceded by at least one conditional prefix-literal (which may be null), possibly followed by one or more other conditional prefix-literals, vertical and horizontal spacing commands, mode commands and/or escape commands. Dummy field selectors may not have suffix-literals. A repeatable prefix-literal is allowed.

A dummy field selector may be used to avoid the unexpected blank lines which may be output in the presence of unattached spacing commands. The required spacing command is attached to a dummy field selector as a prefix-literal and preceded by a null literal, eg
`""CnDt` [where t is field tag for a previously formatted field]

Some examples of these commands are given below:

Format	Output
`"[Only in English]"n76`	`[Only in English]`
`"(Anon.)"n70,v70+|; |`	`Grieve, B.J.; Went, F.W.`
`"(Anon.)",n80,v80+|; |`	`(Anon.)`
`"[Conference paper]"d44`	`[Conference paper]`
`"[no date]" n26^c,v26^c`	`1965`
`"[no date]"n27^c,v27^c`	`[no date]`

F. Expressions

The formatting language allows you to evaluate values and/or compare values through the use of ***expressions.*** Expressions are constructs that, when executed, return a value. This value may be a string of characters (e.g. the contents of a given field or a literal), in which case the expression is called a ***string expression;*** a number, in which case the expression is called ***numerical***; or it may be a truth value (True or False), in which case the expression is called ***Boolean.*** CDS/ISIS also provides a set of ***functions,*** which, on the basis of arguments you provide, perform a specific process and return a value. Functions returning a number are called ***numerical functions;*** those returning a string of characters are called ***string functions;*** and those returning a truth value are called ***boolean functions***. Only string functions may be used directly as formatting commands. Numerical expressions may be used in boolean expressions or as arguments of functions. Boolean expressions and boolean functions may only be used in the context of an IF command.

1. Numerical expressions

Numerical expressions are formed with operands which have a numerical value and operators specifying the calculations to be performed.
The operands you may use in a numerical expression are as follows:

numerical constants: such as 5 18 98.65; numerical constants may be represented as optionally signed integers, decimal numbers, or in scientific exponent notation, e.g. 1.5E5 (meaning 1.5 times 10 power 5, i.e. 150000);
numerical functions: such as val(v10) (these are described under “Numerical functions” on page 104*);*
MFN: the value of the MFN of a record
numerical expressions: when used as an operand, an expression must be enclosed in parentheses, for example (val(v20)-5).

The available operators are:

+	addition (or unary +);
-	subtraction (or unary -);
*	multiplication;
/	division.

As in normal algebra, in the absence of parentheses, unary operators are executed first and multiplications and divisions are performed before additions and subtractions. A series of two or more operators at the same level are executed from left to right. You may use parentheses to alter this order of evaluation: expressions enclosed in parentheses are evaluated first and, inner parenthetical expressions are evaluated before outer expressions.

Note that, as field selectors (e.g. v10 or v20^a) yield a string of text, they cannot be used as operands in a numerical expression. The VAL function, however, may be used to convert the contents of a field or subfield to a numerical value.

Likewise, a numerical expression cannot be displayed directly but must first be converted to a character string using the F function.
Examples of numerical expressions are given below (where it is assumed that MFN=10, vl^a=10, vl^b=20 and v2=30):

Expression	**Value**
`0.155e+3`	`155`
`1e-3`	`0.001`
`2*3+9`	`15`
`2*(3+9)`	`24`
`10-(4*(2-1))`	`6`
`15*0.001`	`0.015`
`mfn+l00`	`110`
`val(v2)+val(v1^a)*7.5`	`105`
`(val(vl^a)-val(v1^b))/100`	`-0.1`

The IBM PC version of CDS/ISIS converts all numbers used in calculations to single precision floating point numbers. This provides a precision of about seven digits with a maximum value of about 1.701411E38. For greater precision, use a string expression comparison eg v49<='19970421'

2. String expressions

String expressions are formed with operands which are strings of characters. As CDS/ISIS provides no explicit string operator, a string expression always consists of a single operand which may be one of the following:

unconditional literals: such as `'some text'`
field selectors: which may include an ***offset/length*** command (e.g. v26^c*2.2);
string functions: such as `S(v24,v25,v26)` (these are described under “String functions”).

3. Boolean expressions

Boolean expressions are used to determine whether a set of one or more conditions are true or false and evaluate to a truth value. The operands of a boolean expression can be one of the following:

relational expressions: which compare two values and determine whether a given relationship exists (see below), such as `MFN <10`;
boolean functions: such as `p(v24)`, which return a truth value (these are discussed under “Boolean functions”).
Relational expressions allow you determine whether a certain relationship between two values is verified. The general form of a relational expression is:

```
expression-1 relational-operator expression-2
```

where:
Expression-1 is either a numerical or a string expression;
Relational-operator is one of the following:

=	Equals
<>	Not equal to
<	Less than
<=	Less or equal than
>	Greater than
>=	Greater or equal than
:	Contains (may only be used for string expressions)

expression-2 is an expression of the same type as expression-1, i.e. expression-1 and expression-2 must be either both numerical or both string expressions.

The relational operators `= < > < <= > >=` have their normal meaning when applied to numerical expressions (within the limits of the precision of numerical values defined under “Numerical expression”).
When comparing string expressions, the following rules apply:

except for the: (Contains) operator, strings are compared exactly as they occur, i.e. upper case and lower case letters are compared according to their Corresponding ASCII character code (e.g. A will compare *less than* a);

two string expressions are **not considered equal** unless they have the same length. If two expressions yielding strings of different length are such that they are character for character equal up to the length of the shorter one, then the shorter string is considered to be less than the longer one.

The **:** (contains) operator searches a string of characters (defined by expression-2) in another string (defined by expression-1). If the second operand occurs anywhere in the first operand the result is True. This operator is case insensitive:
lower case letters are considered to be equal to the corresponding upper case letter. For example, the result of:

```
v10 : 'chemis'
```

would be True if, and only if, field 10 contains the string ***chemist,*** otherwise the result would be False. Note that the second operand may be any arbitrary string of characters and need not be an actual word. Thus, in the example above, the result would be True not only if field 10 contained the word ***chemist*** but also if it contained ***chemistry, biochemistry, photochemistry,*** etc.
The operands of a Boolean expression may be combined with the following Boolean operators:

NOT this operator produces the value True if its operand is False, and the value False if its operand is True. The NOT operator may only be used as a ***unal*** operator, i.e. it is always applied to the Boolean expression which follows it;

AND this operator produces the value True if *both* operands are True. If either operand is False then the result is False;

OR this operator performs an inclusive-OR operation. The result is True if either or both operands are True, otherwise it is False.

In evaluating Boolean expressions, and in the absence of parentheses, CDS/ISIS will execute NOT operations first, then AND operations before OR operations. Series of two or more operators at the same level are executed from left to right. You may use parentheses to alter this order of evaluation:

expressions enclosed in parentheses are evaluated first and, inner parenthetical expressions are evaluated before outer expressions.

Examples of boolean expressions are given below:

Expression	**Value**
Mfn=4	True
not mfn=4	False
not (not mfn=4)	True
v24='plants'	False
v24: 'plants'	True
v24: 'PLANTS'	True
v44.6='method'	False
v44.6='Method'	True
v24: 'plants' and v44: 'method'	True

G. Functions

A Function evaluates a value (called the ***function value*** or the ***returned value)*** which is then substituted for the function in the calculation of the expression. Functions may have one or more ***arguments,*** which you must supply, the value of which is used in the evaluation of the function value. Thus the value of a function depends on the value of the arguments supplied. These are enclosed in parentheses and separated by commas.
Arguments may be of three types:

format: a CDS/ISIS format, which may contain any legal formatting command; except for the REF function (see "REF(expression,format)"), when a format is used as an argument, it is *the text resulting from its execution which is passed to the function,* rather than the *format itself;*

numerical expression: when a numerical expression is used as an argument it is first evaluated and the value of the expression is then passed to the function;

field selector: a field selector argument may be either a field or a subfield command; it may not contain an offset/length command.

The available functions and the corresponding arguments are described below, classified according to the type of value they return.

1. Numerical functions

a. VAL(format)

The VAL function returns the numerical value of its argument. The argument *format* is a CDS/ISIS format and may contain any legal formatting command. CDS/ISIS executes the argument to produce a string of text. This is then scanned from left to right until a valid numeric value is found (which may be in scientific exponent notation). The VAL function returns this numeric value converted to its internal machine representation, suitable for performing calculations. If no numeric value can be identified a value of zero is returned. If the text contains more than one numerical value only the first one is returned.
For example (assuming that v1^a = 10, v1^b=20 and v2=30):

Format	Value
val('15.79')	15.79
val(v1)	10
val(vl^a)	10
val(v2)	30
val('19',vl^b)	1920
val('xxxx7yyyy8zzzz')	7
val('abs 5.8e-4 ml')	0.00058
val('water')	0
val('Jul-Aug 1985')	0

Note that the last example returns 0 (and not 1985), because CDS/ISIS takes the hyphen between Jul and Aug as the beginning of a negative numerical value and the A of Aug as the end; therefore the value extracted is just '-', which evaluates to 0. It is therefore important to clearly define data entry rules for those fields or subfields which will be used in numerical calculations.

b. RSUM(format)

The RSUM function returns the sum of one or more numerical values. The text produced by the argument is scanned from left to right, as for the VAL function, and *all* the numerical values it contains are added together. The final total is the function value. Individual values must be separated by one or more non numeric characters, and it is your responsibility to insert these through the format supplied as argument. RSUM may be used to compute the sum of the numerical values contained in all the occurrences of a given repeatable field.
For example (assuming that field 1 has four occurrences containing 1, 2, 3 and 4):

Format	Value
rsum('10,20,30')	60
Rsum(v1|;|)	10
Rsum(v1|,|,'48,3.5')	61.5

c. RMIN(format)

The RMIN function returns the minimum value of one or more numerical values. The text produced by the argument is scanned from left to right, as for the VAL function, and *all* the numerical values are extracted. The algebrically smallest of these is the function value. Individual values must be separated by one or more non numeric characters, and it is your responsibility to insert these through the format supplied as argument. RMIN may be used to compute the minimum of the numerical values contained in all the occurrences of a given repeatable field.
For example (assuming that field 1 has four occurrences containing 10, 20, 30 and 40):

Format	Value
rmin('1,2,-3')	-3
rmin(vl|;|)	10
rmin(vl|,|,'48,3.5')	3.5

d. RMAX(format)

The RMAX function returns the maximum value of one or more numerical values. The text produced by the argument is scanned from left to right, as for the VAL function, and *all* the numerical values are extracted. The algebrically largest of these is the function value. Individual values must be separated by one or more non numeric characters, and it is your responsibility to insert these through the format supplied as argument. RMAX may be used to compute the maximum of the numerical values contained in all the occurrences of a given repeatable field. For example (assuming that field 1 has four occurrences containing 10, 20, 30 and 40):

<table>
<tr><th>Format</th><th>Value</th></tr>
<tr><td>rmax('1,2,-3')</td><td>2</td></tr>
<tr><td>rmax(v1|;|)</td><td>40</td></tr>
<tr><td>rmax(vl|,|,'48,3.5')</td><td>48</td></tr>
</table>

e. RAVR(format)

The RAVR function returns the average value (arithmetic mean) of one or more numerical values. The text produced by the argument is scanned from left to right, as for the VAL function, and *all* the numerical values are extracted. The average value is then computed and returned as the function value. Individual values must be separated by one or more non numeric characters, and it is your responsibility to insert these through the format supplied as argument. RAVR may be used to compute the average value of the numerical values contained in all the occurrences of a given repeatable field. For example (assuming that field 1 has four occurrences containing 10, 20, 30 and 40):

<table>
<tr><th>Format</th><th>Value</th></tr>
<tr><td>ravr('1,2,-3')</td><td>0</td></tr>
<tr><td>ravr(vl|; |)</td><td>25</td></tr>
<tr><td>ravr(v1|,|, '48,3.5')</td><td>25.25</td></tr>
</table>

f. L(format)

The L function uses the text produced by the argument as a search term to the inverted file and returns the MFN of the first posting (if any). Before looking up the inverted file the term is automatically converted to upper case. If the term is not found the function value is zero. The L function is normally used in conjunction with the REF function to implement table lookup (see under “REF (expression,format)”, for examples on the use of the L function).

Note that the argument format is executed using the current display mode (see “Mode command”). This is important because *the use of* an incorrect mode may result in not finding the term on the inverted file. As a general rule you should use the same mode used in the inverted file FST.

g. LR((format)[, from, to])

Like the L function, LR searches the inverted file for the term defined by format, and returns **all the postings** of the term. For example:

```
ref(lr((v10)),v1,v2)
```

will retrieve fields 1 and 2 from all the records posted under the term contained in field 10. You may limit the range of postings to be retrieved by using the optional from and to parameters. For example:

```
lr((v10),3,7)
```

will only retrieve postings 3 to 7. The parameters “from” and “to” are optional and must be valid numeric expressions.

h. NPST(format)

Like the **L** function, **NPST** searches the inverted file for the term defined by *format*, and returns the number of postings of the term.

i. NOCC(Vtt)

Return the number of occurrences of field *tt*. For example:

```
f(nocc(v70))
```

j. OCC

The **OCC** function returns the number of the current occurrence within a repeatable group. It may be used to produce numbered lists. For example the format:

```
(v70/)
```

will produce one line for each occurrence in the field 70. The following format:

```
(if p(v70) then f(occ,1,0),'. ' fi,v70/)
```

will produce a numbered list as the following:

```
     1. First Author
     2. Second Author
     3. Third Author
```

k. SIZE(format)

Return the size of the string generated by *format*. Note that this function is mode-sensitive. For example, if field 10 contains 20 characters, then `size(mpl,v10)` will return 20, however `size(mdl,v10)` will return the value 22, counting, therefore, the period and space automatically generated in data mode.

l. TYPE(type, format)

This function tests whether the string generated by *format* is of the type defined by *type* and returns **1** if the string corresponds to the specified type or **0** if it does not correspond. The **TYPE** function has two different forms:

TYPE('pattern',format); or
TYPE(numerical expression,format).

The first form may be used to test if the string corresponds to a certain pattern. For example:

```
type('XXA-99-99-99',v10)
```

will return 1 if field 10 corresponds to the pattern or 0 otherwise.
The second form may be used to test other conditions according to the value of *numerical expression*, which must be one of the following:

1 - alphanumeric (the string contains only alphabetic or numeric characters);
2 - alphabetic (the string contains only alphabetic characters);
3 - numeric (the string contains only numeric characters);
4 - decimal integer (the string is an optionally signed integer, e.g. -24)
5 - decimal number (the string is a numeric value, including scientific notation).

For example:

```
type(3,v40)
```

will return 1 if field 40 contains only the digits 0-9.
Note that CDS/ISIS will use the table ISISAC.TAB to determine whether a character is alphabetic or not.

m. TAG

This command works only in a repeatable group and it is meant listing the field tags in the record in their order of insertion. Repeatable tags will list all occurrences. Syntax: tag (returns a numeric value). This command works in all Winisis versions since 1997.

Example:

```
(if p(v0) then f(tag,0,0),| |,v0/,fi)
```

the above will list all tags of the current record with their content:

```
100 ^cTRINIDAD AND TOBAGO^t(1809)66-00000
200 <public><non-profit>
250 <training><conference-organization>
325 <Caribbean Area>
350 <political science>
```

2. String functions

String functions may be used both as operands of string expressions and as formatting commands. When used as a command, the function value will be formatted as if it was a field in the record.

a. F(expr-1 ,expr-2,expr-3)

The F function converts a numeric value from its internal floating point representation to a character string. The three arguments are all numerical expressions. The first argument, ***expr-1,*** is the number to be converted. The second argument, ***expr-2*** is the minimum output width and the third argument ***expr-3*** is the number of decimal places.

The second and third arguments are optional, but do not omit them if a well defined output is required. Note, however, that ***expr-2*** cannot be omitted if ***expr-3*** is present.

expr-2 gives the *minimum* width, i.e. the function value will be a character string of at least ***expr-2*** characters and, if the converted numerical value requires ***expr-2*** characters or less, it will be right adjusted within this width. If the number of characters required to represent the value of ***expr-1*** is greater than the width given, then CDS/ISIS will use additional character positions as needed. In this case the output string will be longer than ***expr-2*** characters.

expr-3 defines the number of decimal places. If missing, the result may be in scientific exponent notation and, if also ***expr-2*** is missing a default width will be used. If present, the result will be a rounded fixed point representation of ***expr-1,*** with ***expr-3*** digits after the decimal point. If ***expr-3*** is zero then ***expr-1*** is first rounded toward the nearest integer and output as an integer value with no decimal point.

For fixed point and integer conversion, if the integer part of the number is too large to be represented, the output is replaced by a series of asterisks (*).

The F function may be used to align a column of numbers on the decimal point by choosing an appropriate with.

Examples of the F function are given below.

Format	**Value**
F(1)	1.000000e+00
f(1,10)	1.000000
F(-l,10,2)	-1.00
f(1,5,2)	1.00
F(1,8,2)	1.00
f(mfn,1,0)	4
F(mfn,2,0)	4
F(mfn,3,0)	4

b. REF(expression format)

The REF function allows you to extract data from an alternate master file record. The first argument is a numerical expression giving the MFN of the alternate record to be selected and the second argument is the format to be applied to this record. If the value of ***expression*** does not correspond to the MFN of an existing record in the data base, then REF will produce a *null string* (i.e. no output). The process performed by the REF function is represented below, where it is assumed that the current record being formatted is record 1.

As you can see from this example, the REF function is a very powerful device, as it allows you to gather together data which is in fact stored in *different* records of the data base, and make it appear to the user as if stored in the same record. In this first example, records 1 and 98 are linked by specifying in field 4 of record 1 the MFN of the related record, which contains the name of the country in English and French: through your format you may then select either language, by simply specifying the relevant tag in the related record.

In certain cases, the linking of records by means of the MFN may be inconvenient from the point of view of entering the data. Beside the fact that any typing error in the MFN of the related record will result in displaying wrong data, it may require time to determine the correct MFN to use. In the case shown in the figure below, for example, it may well be that the source document from which the data is entered already contains a normalized country code (‘UK’ in our case). Using the MFN as a link to the country record, may therefore require the consultation of listings or a search in the data base to find out that the MFN of the record corresponding to ‘UK’ is 98: it would be much more convenient to be able to enter ‘UK’ rather than ‘98’. This is in fact possible and you can obtain the same output given in Figure 56, by organizing the data base in such a way that you may take advantage of the L function .

The **L function** finds the MFN corresponding to a search term. You may use it therefore to convert a character string (such as ‘UK’) to an MFN. In order to be able to use the L function you must establish a ***unique*** relationship between a given character string and its corresponding MFN. The Inverted file provides such a mechanism (see “Inverted file”). In our example, therefore, it would be sufficient to *invert* field 10 of the ‘country’ records to establish a unique relationship between the country code and the corresponding MFN (note that the concept of *uniqueness* is important as the L function assumes that the key it is searching for has one and only one posting. It is your responsibility to make this relationship unique by using, if necessary, a search term prefix as indicated under “Inverted file FST”). The figure below illustrates this technique. It is assumed here that field 10 of the ‘country’ records is inverted with the prefix ‘CC=’.

CDS/ISIS makes no assumption as to the nature of a relationship existing between two records. It simply provides a mechanism for linking records. A particular implementation would normally convey to the user the *meaning* of a relationship through an appropriate use of the formatting language and a specific data base design. For example if a bibliographic record must be linked to a supplier record and to a borrower record, you should use two different fields to store the link to the supplier and to the borrower in order to reflect the different nature of these relationships.

Note, that furthermore, because the second argument of the REF function is a format, it is possible to use this function in a recursive manner, to establish hierarchical relationships of higher orders, such as those that would be required to display the hierarchical relationships in a Thesaurus.

As many REF function references as wanted may be used in a format[5], provided that the output limit (i.e. the size of the work area) is not exceeded. This is currently limited to 65000 characters. Note that this limitation is an *implementation restriction* rather than a restriction of the formatting language.

[5] Much depends from the available memory. More REF instructions also slow the processing of the information and the display of the record. In particular, when making inverted files using REF, Winisis may fail with large amount of data.

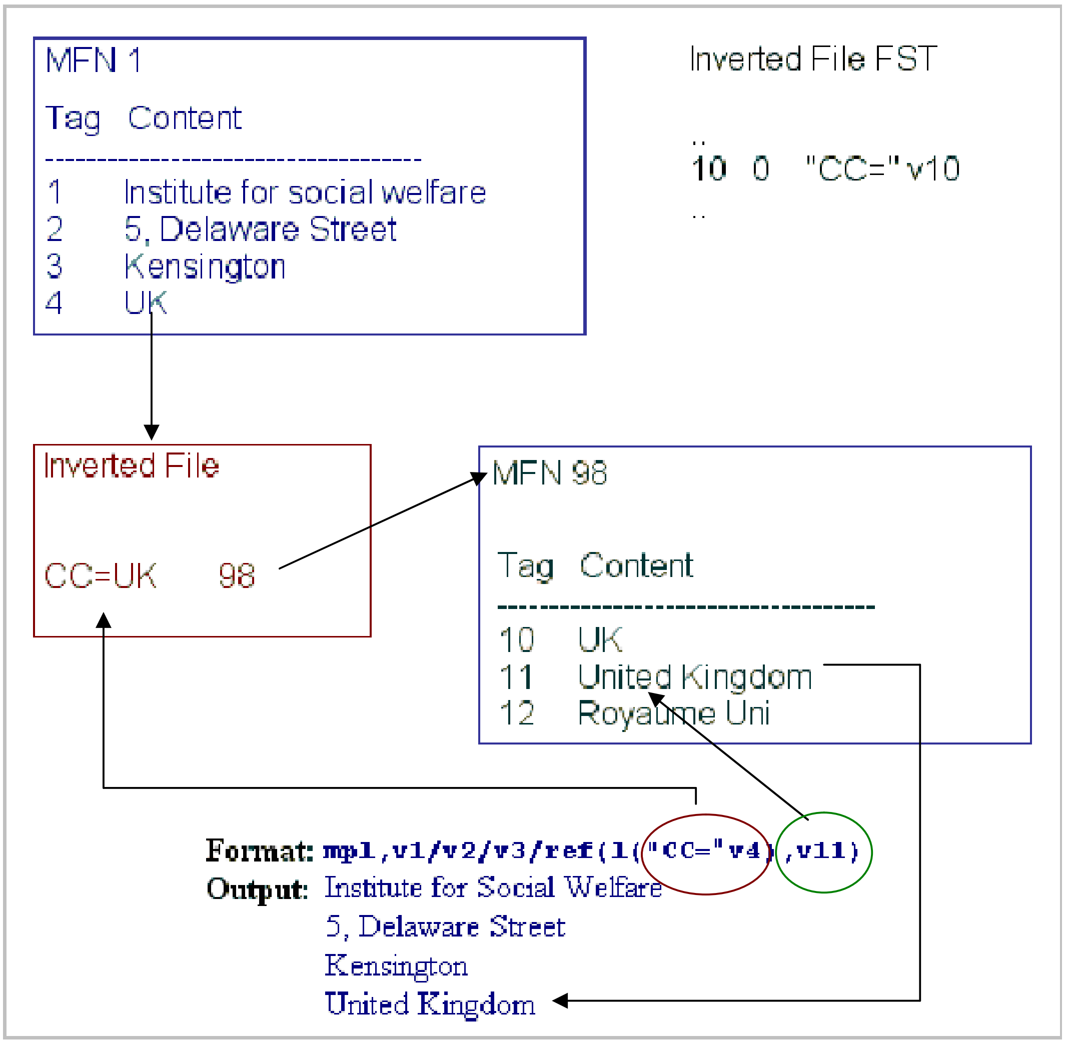


Figure 56 - REF example

c. Alternate data base

The **REF**, **L**, **LR** and **NPST** functions may specify a data base qualifier to refer an alternate data base. When the application is so designed, the data base specified in the **REF** function may be different from the one specified in the **L** or **LR** functions. The data base qualifier is specified as follows:

```
->dbref
```

where *dbref* is the name of the alternate data base (which must be in the data base path specified in parameter 5 of SYSPAR.PAR or for which there is a dbn.PAR in the data base path).
For example:

```
ref->bib(l->book(v10),v100,v200)
```

In this example, supposing the current data base is CDS.MST, the different parts of the format will be executed as follows:

Format piece	**Database**
`Ref->bib(..)`	CDS
`l->book(..),v100,v200)`	BIB
`V10`	CDS

BOOK's MFN corresponding to term v10 of CDS is used as reference for the REF function that is retrieving the content of BIB's v100 and v200.

More realistically, you may use a format such as below (the single quotes are optional):

```
ref->'item'(lr->'item'((|CN=|v37)),v100," "v200/)
```

v37 is a field in the current database CDS containing the class number. The above format looks up the inverted file of the ITEM database for CN=.. (which has multiple postings). It retrieves the MFN of each posting in ITEM, opens each record, and displays the contents of fields v100 and v200 on separate lines.
Note that the use of a repeatable group is mandatory with the **LR** function (hence the double parentheses).

d. S(format)

The S function returns the text produced by its argument. As mentioned earlier CDS/ISIS provides no explicit operators for string expressions. The S function may be used, however, to implement string concatenation. It is particularly useful in boolean expressions, where it may be used to implement an *implicit* OR, which is more efficient (and more concise) than using an explicit OR operator. For example, the two following boolean expressions:

```
S(mdl,vl0,v20,v30) : 'water'

V10 :'water' or v20 : 'water' or v30 : 'water'
```

are equivalent (they are both True if any of the fields 10, 20 or 30 contain the string 'water'), but the first will execute faster than the second.
An *indentation command* can be added to provide indentation for a group of fields, e.g.

```
S(v10,v20,v30)(3,3)
```

e. Substring: SS(pos, length, format)

A substring of a string can be produced in two different ways:

1. by using the ***offset.length** construct with the **S** function, as in the following example: **`S(v24,v69)*3.5`** (in this case CDS/ISIS will extract 5 characters starting from the 4th position of the string returned by **S**);
2. by using the substring function ***SS(pos,len,format).*** That function will take the substring of the string returned by *format* beginning at position *pos* and *len* characters long. For example **`SS(1,5,v30)`** will extract the first 5 characters of field 30.

The main difference between the two forms is that in the **SS** function both *pos* and *len* can be numeric expressions, while in the ***offset.length** construct the values must be numeric constants.
Note also that * works with an offset (starting from 0), while SS works with a position (starting from 1).

f. DATE(exp)

Returns the current date and/or time in the format specified by the numeric expression *exp*. The value of *exp* can be one of the following:

1 - returns a date stamp identical to the one returned by the DATESTAMP function of ISIS Pascal, i.e. an 18-byte string of the form MM-DD-YY HH:MM:SS (e.g. date(1) could return: 09-30-97 15:03:44);

2 - returns only the date (e.g. date(2) could return: 09-30-97);

3 - returns only the time (e.g. date(3) could return: 15:03:44).

g. DB

The **DB** function returns the name of the current data base.

h. Format exits

In a format you may invoke CDS/ISIS Pascal programs you have written to perform special formatting functions required by a particular application, which could not otherwise be obtained by using the formatting language. These programs are called ***Format exits.*** As format exits are developed to satisfy specific needs, their description is beyond the scope of the formatting language. CDS/ISIS provides, however, a normalized way to interface format exits with the formatting language.

From the point of view of the formatting language a Format exit is a string function with a ***format*** argument. The argument is first executed and its output is passed to the function. A format exit returns a character string which CDS/ISIS handles as if it was a field in the record being formatted.

From the point of view of CDS/ISIS Pascal a Format exit is a program declared with the [FORMAT] attribute (see the ***CDS/ISIS Pascal*** manual for details on the interface). Before a Format exit can be referenced in a format, the corresponding program must have been compiled successfully.

A Format exit is invoked as follows:

&Name(format)

Where:

& identifies this as a Format exit invocation;
Name is the name of the CDS/ISIS Pascal program to be executed;
Format is the argument.
An indentation command (see under "Indentation command") may optionally be appended to a Format exit invocation, which CDS/ISIS will apply to the output string. Below is an example of a Format exit, which simply returns the argument as the function value.

Format exit

```
Program SAMPLE(arg: string; lw,occ: real; str: string) [FORMAT];
begin
Str:=arg;
end.
```

Format	**Output**
&sample('xxx')	xxx
&sample(v26^a)	Paris
&sample(mhu,v24)(0,5)	AN ELECTRIC HYGROMETER APPARATUS FOR MEASURING WATER-VAPOUR LOSS FROM PLANTS IN THE FIELD

Nowadays, however, it is recommended to use ISIS_DLL instead of Isis Pascal.

3. Boolean functions

a. P(field selector)

The P function returns True if the record being formatted contains at least one occurrence of the field or subfield indicated by the argument. For example:

Format	**Value**
p(v24)	True
p(v26^d)	False
p(v70[2])	True
p(v80)	False

b. A(field selector)

The A function returns True if the record being formatted contains no occurrence of the field or subfield indicated by the argument.
Note that the absence of a field implies the absence of all its subfields.
Therefore, if the field selector specifies a subfield the A function returns True if either the field is present, but the specified subfield is absent, or the field itself is absent. For example:

Format	Value
`a(v24)`	`False`
`a(v24^s)`	`True`
`a(v26^d)`	`True`
`a(v80)`	`True`

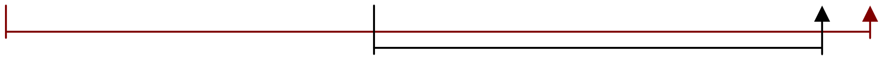
H. IF command

The IF command allows you to implement *context-sensitive* formats, i.e. formats able to produce output which may vary depending on the *contents* of the record being formatted. It is coded as follows:

IF *condition* THEN *format-1* ELSE *format-2* FI
where:
Condition is a boolean expression as defined under "Boolean expressions";
Format-1 is a CDS/ISIS format which will be executed if and only if the boolean expression is True;
Format-2 is a CDS/ISIS format which will be executed if and only if the Boolean expression is False.

The ELSE *format-2* clause is optional and may be omitted. The IF, THEN and FI keywords are always required, although *format-1* may be omitted when an ELSE clause follows (e.g. whenever nothing has to be output if *condition* is True). An IF command may therefore also take one of the following alternate forms:
IF *condition* THEN *format-1* FI
IF *condition* THEN ELSE *format-2* FI
As there is no restriction in the commands you may use in *format-1* or f*ormat-2*, IF commands may be nested to any desired depth. The FI keyword must, in this case, be used to *close* each IF command (you may think of IF and FI as a pair of parentheses). For example:

```
if p(vl) then v24 else if p(v2) and a(v3) then v5 fi fi
```



The IF command is particularly useful to develop generalized formats for integrated data bases, which contain different types of records. In this case you will normally have distinctive mark for each type of record (typically there will be a field containing a code identifying the type of record). Thus, by checking the type of record with an IF command, you may perform, through a single format, specific formatting for each type.

I. Repeatable groups

A repeatable group consists of a set of formatting commands enclosed in parentheses. The meaning of each command is the same as described above, except that repeatable fields are handled in a special way.
In order to understand the concept of repeatable groups you should first know how CDS/ISIS handles repeatable fields. In the absence of any other indication, CDS/ISIS treats all the occurrences of a repeatable field (in the order they have been entered) as a *single string of text.*
A repeatable group alters the way CDS/ISIS would normally handle the occurrences of a repeatable field, by processing one occurrence at a time rather than all together. This process is described below, with some examples.
When CDS/ISIS encounters the open parentheses of a repeatable group it proceeds as follows:

1. An *occurrence counter* is initialized to 1.
2. The format enclosed in parentheses is then executed in such a way that all field selectors within the group will only output the occurrence of the field corresponding to the current *occurrence counter.*
3. If no output was produced (i.e. if there were no more occurrences of any of the repeatable fields referenced within the group), then the processing of the repeatable group is terminated. Otherwise the *occurrence counter* is increased by 1 and steps 2 and 3 are repeated.

Note that *all* formatting commands within a repeatable group are processed one occurrence at a time (as explained above), including, therefore, fields referenced in IF commands, expressions and functions, as well as in string functions used as commands.
Because of the processing explained above, you should not use unconditional literals within a repeatable group (if you do, they will be output one time more than you would expect).
In most cases the use of simple formatting commands, such as the mode command or repeatable literals, is sufficient to adequately handle repeatable fields, as in the examples given below:

Format	Output
`mpl,v70`	`Grieve, B.J.Went, F.W.`
`mdl,v70`	`Grieve, B.J.  Went, F.W.`
`v70+|; |`	`Grieve, B.J.; Went, F.W.`

There are cases, however, where you will need to format repeatable fields in other ways. A frequent case is, for example, the need to format each occurrence on a new line, which may only be done by using a repeatable group, as shown below:

Format	Output

`v70/v26^a`	`Grieve, B.J.Went, F.W.`
	`Paris`

`(v70/),v26^a`	`Grieve, B.J.`
	`Went, F.W.`
	`Paris`

In the first case the newline command (/) is executed *after* formatting *all* the occurrences of field 70; whereas, in the second case, it is executed after *each* occurrence.
The example below shows the handling of repeatable subfielded fields (assuming that the record contains two occurrences of field 20 as indicated). Here the use of a repeatable group has helped to properly display the various subfields of *each* occurrence of the repeatable field in a tabular manner.

Field	Record content
20	`^aNew York^bMcGraw Hill^cl988`
20	`^aLondon^bAcademic Press^cl975`

Format: `/(v20^a,cl3,v20^b,c31,v20^c/)`
Output:

```
      New York    McGraw Hill        1988
      London      Academic Press     1975
```

Format: `/v20^a,c13,v20^b,c31,v20^c/`
Output:

```
      New YorkLondon
                  McGraw HillAcademic Press
                                     19881975
```

If you need to output a literal before the data produced by a repeatable group, you may use an unconditional literal or a conditional literal. Note, however, that if you use a conditional literal it must be

associated with a field selector (a repeatable group is not a field selector); you must use a dummy field selector for this purpose (see below).
As a further example of a repeatable group, assume that in a personal history record field 10 contains the previous employers of a person and field 20 contains the functions that the person had when working for a particular employer. In such a record, both field 10 and 20 would be repeatable, since a person might have worked for more than one employer. This is a case where a logical relationship exists between two repeatable fields. Below is an example of use of a repeatable group to display these two fields (it also illustrates the use of a dummy field selector).

Record contents			
10	Bedford and Associates	20	Junior programmer
10	Van Allen Inc.	20	System programmer
10	Michigan University	20	Lecturer in Computer Science

Format: `"Employment History"/#d10,(v10(6,6)/v20(12,12)/#)`

Output:
```
Employment History

      Bedford and Associates
            Junior programmer

      Van Allen Inc.
            System programmer

      Michigan University
            Lecturer in Computer Science
```

Format: `"Employment History"/d10,(c7,v10|: |,c37,v20/)`

Output:
```
Employment History
      Bedford and Associates:       Junior programmer
      Van Allen Inc.:               System programmer
      Michigan University:          Lecturer in Computer Science
```

Repeatable groups cannot be nested (i.e. a repeatable group may not contain another repeatable group), unless the inner group is contained in the format argument of a REF function. Thus, for example, the following is a valid format:

```
(v10,ref(val(v20),v10,(v20,v30)))
```

whereas the following is invalid and will produce an error message:

```
(V10, (v20,V30))
```

Note that the use of a repeatable group is mandatory whenever:
1. you use a repeatable field as the argument of the L function;
2. the first argument of the REF function references a repeatable field.

You should also consider whether the use of a repeatable group would be called for whenever a repeatable field is used in the Boolean expression of an IF command.

J. Format errors

While interpreting and executing a format, CDS/ISIS also performs a syntax analysis of the format to ensure that it conforms to the formatting language rules. Whenever CDS/ISIS detects an error in the format, it interrupts the formatting, and issues the message *** Format Error n (where n is an error code). For display or print formats, any output produced before the error is detected will appear before the message. This, together with the error message, will help you in determining the erroneous part of the format.

The possible error codes are as follows:

1	End of format detected while processing a repeatable group. Probably the closing parenthesis delimiting the group is missing
2	Nested repeatable group (i.e. repeatable group inside a repeatable group)
8	IF command without THEN
19	Unmatched (
20	Unmatched). It may also be caused by an invalid operand in an expression
26	The two operands of an operator are of different types (e.g. trying to add a string operand to a number)
28	The first argument supplied to the REF function is not a numerical expression
51	Too many literals and/or conditional commands associated with a field selector
53	IF command not terminated by FI
54	+ sign out of context: CDS/ISIS was expecting a repeatable literal following the + sign
55	Unmatched FI
56	Work area overflow: your format produced too much output which CDS/ISIS cannot handle. The work area size is limited to 65000 characters.
58	One or more arguments supplied to the F function are not numerical expressions
60	A non string function used as command (only string functions may be used as commands)
61	The argument supplied to the A or P function is not a field selector
99	Unknown command (e.g. a misspelled function name or command); may also be due to a missing closing literal delimiter
101	Stack overflow (probably due to a too complex expression)
102	Stack underflow (it may be due to unmatched (. If your format is correct it indicates a CDS/ISIS software problem)

K. Including an external format

You may include an external format in a format by using the ***@name*** function, where *name* is the name of the format to be included. This format must be in the data base path (as specified in parameter 5 of SYSPAR.PAR or parameter 10 of dbn.PAR). For example:

```
if v1=‘BIB’ then @fmt1 else @fmt2 fi
```

In this example, the contents of field 1 will determine which of fmt1 or fmt2 will be executed.

L. Format variables

CDS/ISIS predefines ten (10) numeric and ten string format variables which you may use in your format as applicable. The ten numeric variables are named E0 through E9 and the ten string variables are named S0 through S9. The numeric variables are initialized to 0, while the string variables are initialized to null strings, each time a format is executed.

You may assign or change the value of a numeric variable as follows:

E*n*:=*numeric expression* (for example: **e1:=val(v10)+5**)

and you may assign or change the value of a string variable as follows:

S*n*:=(*format*) (for example: **s5:=(v10)**).

Note that the parentheses around *format* are required.
A numeric variable may be used anywhere a numeric value can be used, e.g. as operand of a numeric expression as in **if e1+10<25 then ... fi**. As any other numeric value, a numeric variable cannot be directly displayed, but must first be converted using the **F** function.
A string variable may be used both as operand of a string expression and as a formatting command.

M. WHILE command

The **WHILE** command provides looping capabilities so that you can repeatedly execute a format. It is coded as follows:
WHILE *condition* (*format*)
where:

condition is a Boolean expression as defined on p. 55 of the CDS/ISIS *Reference Manual*;
format is the CDS/ISIS format to be repeatedly executed while the Boolean expression is **True**.

If the initial value of ***condition*** is **False** then ***format*** will not be executed at all. For the loop to end you must provide inside "***format***" whatever commands are necessary to render "***condition***" **False** whenever the loop must be terminated.
If an infinite loop is generated, Winisis will not respond to the user. For example:

```
e1:=1,e2:=nocc(v70), while e1<=e2 (f(e1,1,0),'. ',v70[e1]/ e1:=e1+1)
```

The example above displays each occurrence of field 70 on a new line preceded by the number of the occurrence, e.g.:

```
     1. First Author
     2. Second Author
     3. Third Author
```

A more complex example is given below.

```
s1:=(v69),e0:=size(s1),e1:=1,e3:=1,
while e1<e0
(
while e1<e0 and ss(e1,1,s1)<>'<' (e1:=e1+1)
e2:=e1+1,
while e2<=e0 and ss(e2,1,s1)<>'>' (e2:=e2+1),
s2:=(ss(e1+1,e2-e1-1,s1)),
if size(s2)>0 then f(e3,1,0),'. ',s2/ e3:=e3+1 fi,
e1:=e2+1)
```

In this example, we scan field 69 for the occurrences of keywords enclosed in < >, and display each keyword preceded by its sequence number, e.g.:

```
     1. First Keyword
     2. Second Keyword
     3. Third Keyword
```

N. The Windows graphic environment

Microsoft Windows, as well as most graphic environments, offers a rich array of text-writing capabilities. For example Windows lets you choose the font[6] to be used for text output.

CDS/ISIS for Windows provides new commands to manipulate the display using various fonts, including proportional fonts, and various font sizes and colors. You can add these commands to your existing formats, in order to enhance the typographical quality of both your screen and printed outputs.

Your existing formats are displayed using by default the fixed, non proportional font "Courier New", to simulate the MS-DOS character-based screen under Windows and to correctly interpret indentation values expressed in number of characters. However, when using proportional fonts, a statement like: "Indent first line by 5 characters" would have no meaning, as each character has a different width. The new commands will help you in designing formats which will greatly improve your output.

The next paragraphs describe in detail these new commands.

1. FONTS command

The **FONTS** command defines the set of fonts (also called the *font table*) to be used in your format. If present, it must be the first command in the format. If absent, CDS/ISIS will use the one specified in parameter 110 of SYSPAR.PAR or the default Windows font table if parameter 110 is also missing.

The **FONTS** command is coded as follows:

FONTS ((*family1,font1*) , (*family2,font2*), . . .)

where:

***family*n** is the font family, and
***font*n** is the name of the font.

A font is defined by a *font family* and a *font name*, separated by a comma and enclosed in parentheses. The font family is a group of fonts having similar characteristics, the font name is a specific font in a particular family. While font family names are pre-defined (see the table below), font names depend on the font supplier and the actual fonts available in your Windows environment (you may obtain the list of font names installed on your computer by using the **Fonts** command of Windows **Control Panel**).

The following table defines the font families you may use and gives examples of font names for each family.

Font family	Description	Examples
nil	Unknown or default fonts (default)	
roman	Roman, proportionally spaced serif fonts.	Times New Roman, Palatino
swiss	Swiss, proportionally spaced sans serif fonts.	Arial
modern	Fixed-pitch serif and sans serif fonts	Courier New, Pica
script	Script fonts	Cursive
decor	Decorative fonts	Old English, ITC Zapf Chancery
tech	Technical, symbol, and mathematical fonts.	Symbol

CDS/ISIS numbers each font you define in the font table starting from 0. For example, if you wanted to use the “Times New Roman” and the “Arial” fonts you would use the following command:

```
fonts((roman,Times New Roman),(swiss,Arial))
```

[6] A font is a collection of characters that have a unique combination of height, width, typeface, character set, and other attributes. An application uses fonts to display or print text of various faces or sizes. For example, word processing applications use fonts to give the user a "*what you see is what you get*" (WYSIWYG) interface.

In this case the “Times New Roman” font will be font 0 and the “Arial” font will be font 1. Later in your format you may select either font by using the **F*n*** command (described below) to activate the desired one.

2. COLS command

The **COLS** command defines the set of colors (also called the *color table*) to be used in your format. You may only use a **COLS** command immediately following a **FONTS** command. In other words, in order to use the **COLS** command, this must be preceded by a **FONTS** command. If you provide no **COLS** command, CDS/ISIS will use the color table specified in parameter 110 of SYSPAR.PAR or the default Windows color table if parameter 110 is missing.

The **COLS** command is coded as follows:

COLS ((*red1,green1,blue1*) , (*red2,green2,blue2*), . . .)

where:

***red*n** is a number from 0 to 255 indicating the Red index

***green*n** is a number from 0 to 255 indicating the Green index

***blue*n** is a number from 0 to 255 indicating the Blue index

A color is made from a mixture of the three basic colors: red, green and blue, the color index indicating the amount of that color in a scale from 0 to 255. The color palette varies from black (0,0,0) to white (255,255,255). For example the color (255,0,0) indicates a pure red.

CDS/ISIS numbers each color you define in the color table starting from 0. For example, if you wanted to use the red, green and blue colors you would use the following command:

```
COLS ( (255,0,0),(0,255,0),(0,0,255) )
```

In this case, red will be color 0, green color 1 and blue color 2. Later in your format you may select the desired color by using the **CL*n*** command (see below).

3. Paragraph formatting commands

This set of commands concern the alignment of paragraphs (a paragraph being a block of text terminated by a new line).

a. Indentation

You may use the **M** command to set a global left margin as follows:

M (*indent* , *flindent*)

where:

indent is the indentation of the paragraph from the left margin, and

flindent is the indentation of the first line of the paragraph.

Both measures are expressed in *twips* (there are 1440 *twips* in one inch, and 567 *twips* in a centimeter).

This command is similar to the indentation command following a field selector (e.g. V10(5,10)), but it allows you not only to control more precisely the indentation, but also to apply it to more than one field.

b. Tabulation command

You may use the **TAB** command to tabulate to a desired position of the line, as follows:

TAB or

TAB (*value*)

where:

value is the position of the next character expressed in *twips* from the current left margin.

If ***(value)*** is missing, a default value will be taken. This command is similar to the **C** command, but it allows you to control more precisely the tabulation value.

c. Center command

This command allows you to center text between the left and the right margin. It is coded as follows:
QC

d. Justification commands

QJ this command allows you to justify text between the left and the right margin. It is coded as follows:
QR set current paragraph's alignment to right-justification.

Example: `v24/,qr,v69`

e. BOX command

This command allows you to draw a box around a paragraph. The **BOX** command has two forms, as follows:
BOX
BOX(n)
The first form simply draws a border around the paragraph using the current color (see below the **CL*n*** command under *Character formatting commands*). The second form draws a box around the paragraph, using the current color, and paints the background of the box with color ***n*** (as defined in the color table). Note that in this last case the current color must be different from ***n***, otherwise the text inside the box will not be visible, as it will be drawn using the same color as the background color.

f. NP (new page) command

Use NP to break the current page and open a new one. It is possible to use many NP commands on the same format. NP has no effect on the screen.

g. PICT

PICT is used for displaying pictures on the screen. Syntax: **Pict(fmt1[, fmt2])** where "fmt1" and "fmt2" should return an image filename.
Image "Fmt2" is optional. That image will only appear when the mouse passes over the first image specified by Fmt1.
Note that the default path for images is defined in the SYSPAR.PAR. Example (supposing v45 is a filename):

```
"My image: ",d45,pict(v45)/
```

Winisis supports BMP and JPG formats.

h. BPICT

It is a **non**-standard format function setting the current background pattern and/or image.

Syntax 1: `BPICT([(fmtPatt),](fmtImage),style)`

Syntax 2: `BPICT((fmtPatt)[,(fmtImage),style)]`

Style sets the background image alignment and/or scaling. Available settings are:
1 Tile
2 Scaled
4 Centered horizontally
8 Centered vertically
16 Right aligned

32 Bottom aligned

Settings may be combined to get different output styles. For example, use **12** to centre your bitmap in the window (**8**+**4**)

Example:

```
bpict(('pat016.bmp'),('unesco.jpg'),12)
```

i. TITLE

Title is a non-standard format function that assigns a title to the current window.

Syntax: `title(format)`

Example: title(v24) will use the value of field 24 as the database window's title.

j. KEEPL

It is a non-standard format command by which the current paragraph will printed/displayed on a single line regardless of the window/page margins or size.

Example: `keepl,v24,/v69`

4. Character formatting commands

This set of commands concerns the appearance (font, size, color or style) of text characters. They are coded as follows:

Command	**Function**
b	Bold
i	Italic
ul	Continuous underline
f*n*	Font (where ***n*** is the font defined in the font table)
fs*n*	Font size (where ***n*** is the size in half points, e.g. **fs24** defines a size of 12 points)
cl*n*	Color (where ***n*** is the color defined in the color table)

5. Applying formatting to specific objects only

All the above commands take effect immediately. Thus, if you turn bold on, using the **b** command, all the rest of your format output will appear in bold face. You may, however, control the applicability of a command by using braces ({}) to delimit its validity. A left brace ({) signals that all subsequent text attributes will only be applied until the corresponding right brace (}), which restores the attributes that were active before the left brace. In the following format, for example, the **m**, **f**, **fs** and **b** commands will only be applied to field 24:

```
v10,v20/{m(1000,0),f2,fs28,b,v24}/v30
```

6. Adding Hypertext links to formats: the LINK command

The **LINK** command allows you to add interactivity to your format, by establishing a relationship between a field (or set of fields) of a record and an action to be performed. The general format of the **LINK** command is as follows:

LINK((*descriptor*),*action*)

where:

descriptor is a format describing to the user the action to be taken; the output of this format is displayed using color 2 (normally green, by default) and underlined; this text can be clicked with the mouse; note that this format *must* be enclosed in parentheses;

action is a format telling CDS/ISIS the action to be performed; the output of this format is not displayed and must be one of the hypertext commands listed below, which will be executed whenever the user clicks on the item.

The following paragraphs describe the hypertext commands which may be used in the ***action*** part of the **LINK** command.

Note: all hypertext commands contained in ***action*** *must be written in uppercase and must be followed by a space*. Furthermore, the following special characters contained in any literal *must* be coded as indicated below:

Character	**Code as**
\	\\
{	**{**
}	**}**

Also note that both formats (***descriptor*** and ***action***) are mode sensitive. You may therefore have to change the current mode to obtain the desired result.

a. OPENFILE command:

This command let Winisis to automatically find the proper application to open the specified file, if any installed on your computer.

Syntax: link((‘Click to open’),‘**OPENFILE** c:\\mypage.doc′)

If an application on your computer is associated with a DOC documents (for instance MS-Word), the command will open it to show the file mypage.htm

Replaces in many cases the command CMD and can be used in **menu options as well**.

You can also open a web address:

```
link(('UNESCO'), 'OPENFILE http://www.unesco.org')#
```

or open your favourite mail software to write an e.mail:

```
link(('Write'),'OPENFILE mailto:j.smith@provider.net')#
```

or open any document on a shared network directory:

```
link(('Documentation'), 'OPENFILE \\computer-1\ Public\file1.pdf')#
```

c. CMD command

NOTE: it is recommended to use OPENFILE wherever possible.

Runs the Windows or MS-DOS program indicated by command (just like the "File-Run" menu command in the Program Manager). command may include appropriate parameters for the program to be run. Note that for Windows to find the program its path must be either listed in the PATH statement of the autoexec.bat file or you must include it in command. Some examples follow:

Example 1:

```
mpl,link(('Click here to see full text'),'CMD winword',v10)
```

The screen output of this command will simply be:

```
        Click here to see full text
```

however, if the user clicks on this item, CDS/ISIS will call (**CMD**) Word for Windows (**winword**) and display the document whose name is in field 10 (**v10**). Of course, if need be, this name must include the full path.

Example 2:

```
mpl,link(('Play movie'),'CMD player 'v10)
```

The screen output of this command will be:

```
        Play movie
```

If the user clicks on this item, CDS/ISIS will call (**CMD**) the multimedia player (**player**) and play the movie whose name is in field 10 (**v10**). Also in this case the file name in field 10 must include a path if needed.

Example 3:

```
mpl,link(('UNESCO WWW'),'CMD netscape 'v10)
```

The screen output of this command will be:

```
        UNESCO WWW
```

If the user clicks on this item, CDS/ISIS will call (**CMD**) Netscape (netscape) and display the home page of the Internet site whose URL is in field 10 (v10). If for example, field 10 contains ‘http://www.unesco.org’, Netscape will display the UNESCO Web home page.

Example 4:

```
mpl,link ((‘Show picture’),’CMD pbrush ‘,v10)
```

The screen output of this command will be:

```
        Show picture
```

If the user clicks on this item, CDS/ISIS will call (**CMD**) Windows Paintbrush (**pbrush**) and display the picture whose name is in field 10 (**v10**). Also in this case the file name in field 10 must include a path if needed.

Example 5:

Here we assume that field 10 in a data base has three subfields: subfield a which contains the label to be displayed, subfield b which contains the program to be executed, and subfield c which contains the file name. For example:

```
^aMadonna with child^bpbrush^c\pictures\raffael.bmp
```

The following link command:

```
mpl,link((v10^a),'CMD ',v10^b,' ',v10^c)
```

would then produce the screen output:

```
        Madonna with child
```

If the user clicks on this item, CDS/ISIS will call (**CMD**) Windows Paintbrush (**v10^b**) and display the picture ‘\pictures\raffael.bmp’ whose name is in field 10 (**v10^c**).

MENU EXIT

CMD can be called from a Winisis menu to open an external application. You should edit your menu file, which is in the MENU directory. For example, English default menus are in file MNENDF.MST. Open it with Winisis and add somewhere the following field:

```
Open                    Notepad[CMD                    notepad.exe]
```

Save the record and reload the file by changing the current Winisis language to English again. You will be able to access the new function.

d. GOTO mfn

Displays the record whose MFN is ***mfn***. ***mfn*** is a format which must produce a string containing a valid MFN for the current data base. For example:

```
link(('Show link'),'GOTO 'v10)
```

The screen output of this command will be:

```
        Show link
```

If the user clicks on this item, CDS/ISIS will display the record whose MFN is given in field 10.

e. LGOTO term

This command displays the record corresponding to the first posting of ***term***. ***term*** is a format which defines the dictionary term to be looked up. For example:

```
link(('test'),'LGOTO water')
```

The screen output of this command will be:

```
        test
```

If the user clicks on this item, CDS/ISIS will display the record corresponding to the first posting of ‘water’. Note that the example above would have the same effect as:

```
link(('test'),'GOTO ',f(l(‘water’),1,0))
```

The difference is that LGOTO performs the lookup only in response to the user click, while GOTO performs it during the execution of the format containing the LINK command. LGOTO will therefore execute faster and avoids unnecessary dictionary lookups.

f. LAGOTO[/nn] term

The “Approximately Lookup Goto” command looks up for a term and goes to the nearest MFN. Syntax and parameter description:

```
LAGOTO term
LAGOTO/nn term
```

term: the term to be searched.
nn the field in which term should be searched

Very useful used in conjunction with PROMPT to obtain a very simple search engine. For example:

```
Link((‘search’),'PROMPT LAGOTO acc')
```

will present the first dictionary term's record beginning by 'acc'.

g. GOBACK

Displays the record which was displayed immediately before the current one. For example, suppose you are looking at record 45 and you then move to record 124, by clicking on a hypertext link. You may then not remember which record you were looking at before. If your application requires it, you may provide this possibility by using the GOBACK command. For example:

```
link(('[Previous]'),'GOBACK ')
```

h. FORMAT format-file-name[,old-text, new-text]

Changes the current display format to the one specified by ***format-name***. ***format-name*** is a format which defines the name of the format to be loaded. Note that although this format needs not to be one of those specified in the FDT, it must be in the same path of the current data base (as specified in parameter 5 of SYSPAR.PAR or parameter 10 of dbn.PAR). For example:

```
link(('Change format'),'FORMAT cds1')
```

The screen output of this command will be:

```
        Change format
```

If the user clicks on this item, CDS/ISIS will re-display the current record using the format ‘cds1’. Once loaded, the format becomes the current format.
In the FORMAT command you may use two additional parameters to modify the loaded format before it is executed (***old-text*** and ***new-text***), as explained below under *Polymorphic formats.*

i. BROWSE database-name[,mfn,format-name]

Opens the data base specified by ***database-name*** in a new data base window and displays the record ***mfn*** using the format ***format-name***. ***mfn*** and ***format-name*** are optional parameters. By default ***mfn*** is 1 and ***format-name*** is the default format of the data base. ***database-name***, ***mfn*** and ***format-name*** are formats producing the required values. For example:

```
link(('Open THES'),'BROWSE thes')
```

This will open the THES data base in a new window and display record 1 using the default THES format.

```
link(('Open CDS'),'BROWSE cds,10,cds1')
```

This will open the CDS data base in a new window and display record 10 using the format CDS1.
You may use the ***mfn*** parameter without specifying ***format-name***. You must however specify the ***mfn*** parameter (or insert a comma) if you want to specify ***format-name***. All the following examples are valid specifications:

```
link(('Open CDS'),'BROWSE cds,10')
link(('Open CDS'),'BROWSE cds,1,cds1')
link(('Open CDS'),'BROWSE cds,,cds1')
```

j. TEXTBOX format

This command lets you display text, resulting from the execution of ***format***, in a separate window. The following example shows the basic form of the command:

```
        link(('Open new window'),'TEXTBOX ',v24)
```

when you click on the Open new window hyperlink, a new window will pop up, displaying the contents of field 24.

i. TEXTBOX window types

You may create TEXTBOX windows in different ways:
1) as independent windows,
2) as data base child windows,
3) as record child windows.

You may specify the window type by setting the corresponding attribute as follows:

TEXTBOX	users may close this window only by clicking its close button or exiting Winisis.
TEXTBOXCHILD	the window is declared as "child" of the current database. Closing the database, will also automatically close all its "child" windows.
TEXTBOXRCHILD	the window is declared as "child" of the current record. Moving to another record, or closing the database, will also automatically close all its "child" windows.

Some examples of TEXTBOX are given below:

```
'TEXTBOX ',v24
'TEXTBOXCHILD ',v24/(v70/)
'TEXTBOXRCHILD ',@cds1
```

Note that the format of the TEXTBOX command is executed *while displaying the record*. Its output therefore is stored in the display work area, whose maximum size is 64000 characters. For example, if your format produces a string as long as 30000 characters, you must ensure that the TEXTBOX format will not produce more than 34000 characters, or an error will occur. Also note that, for implementation reasons, ***format*** may not contain any horizontal or vertical spacing commands or graphic commands such as F, FS, CL, etc. (if it does they will be ignored). If these are required you may use the LOAD keyword as explained below.

ii. Loading format files

To avoid some of the limitations mentioned above, you may append the keyword LOAD to a TEXTBOX command, indicating that the ***format*** parameter is a format *name* rather than an actual format. In this case, the format will not be loaded or executed until the user clicks on the hypertext link. A loaded format, may produce a string of up to 64000 characters and may include all legal formatting commands. The LOAD keyword may be applied to any type of TEXTBOX. For example:

```
link (('Show record'),'TEXTBOXCHILDLOAD cds1')
```

will open a window displaying the text resulting from the execution of format ‘cds1’.
Note the difference between the following two links:

```
link(('Show record'),'TEXTBOX ',@cds1)
link(('Show record'),'TEXTBOXLOAD cds1'
```

the first, executes the format ‘cds1’ while displaying the record; the second statement executes ‘cds1’ only when the user clicks on the hypertext link Show record. As a further example consider the following formats:

Format CDS:

```
if p(v44)
   then link(('Show SERIES 1'/),'TEXTBOXLOAD cds1')
fi,
if p(v44)
   then link(('Show SERIES 2'/),'TEXTBOX ',@cds1)
fi,
```

Format CDS1:

```
{b,fs30,'SERIES: ',|(|V44|) |}
```

In this case, the hyperlink Show SERIES 1 (in format CDS) will display field 44 in bold using a 15 points font size, while Show SERIES 2 will display field 44 using normal characters.
When you use the LOAD keyword you may also supply format replace parameters as in FORMAT command (see below under *Polymorphic formats*).

iii. Displaying pictures

You may display a picture (which, at present, must be a Windows bitmap file with extension BMP) in a TEXTBOX by appending the IMG keyword to the command. For example, the following command:

```
link(('Show picture'),'TEXBOXIMG c:\\winisis\\ab_jun97')
```

will open a new window displaying the file “`ab_jun97.bmp`”
You may use the IMG keyword in conjunction with any TEXTBOX type. Note, however, that the LOAD and IMG keywords are mutually exclusive. For example:

```
link(('Show picture'),'TEXBOXCHILDIMG c:\\winisis\\ab_jun97')
link(('Show picture'),'TEXBOXRCHILDIMG c:\\winisis\\ab_jun97')
```

Pictures larger than 500 x 300 pixels will be scaled to fit such size.

iv. How to Prompt the user before executing a Hypertext command

The PROMPT prefix allows you to query the user before executing a (polymorphic) format at run-time.

The command prefix PROMPT

The command PROMPT can be added prior to any other commands in the Link statement. For example:

```
..,Link(('Test'),'PROMPT TEXTBOX Ciao'),..
```

When clicking the above hypertext, a small textbox appears just beside the clicked term ('Test'). A blinking cursor indicates that Winisis is listening for the user input.
If the focus is moved to a different object, the textbox will disappear.
Use ENTER to validate the input and take out the prompt from the screen.

How to use the user input
Winisis will substitute each $$ (double dollar sign) in the rest of the hypertext command with the user input. For example:

```
..,Link(('Test'),'PROMPT TEXTBOX $$'),..
```

will open a new window containing the string the user typed in.

Consider the following hypertext command:

Base format: TEST.PFT

```
Link(('Click here to search...'), 'TEXTBOXLOAD TESTR')
```

In response to the user click, it will execute a second external format. Such format will search for a given term, opening a new window containing the search results:
Loaded format: TESTR.PFT

```
Ref(l('water'),v24)
```

To make this interactive, put the PROMPT prefix in TEST.PFT:

```
Link(('Click here to search...'), 'PROMPT TEXTBOXLOAD TESTR,zzzz,$$')
```

and change TESTR.PFT as follows:

```
Ref(l('zzzz'),v24)
```

Result: when you click on 'click here to search' you'll get a prompt box; type in your query and press ENTER: a new window will appear with the result.

How it works?
Consider the following:
It is impossible to delay the execution of the current format;
The only way to pass values through formats is by using polymorphic extensions. In other words, the mechanism works by doing a double substitution:
- the $$ is substituted by the user input (e.g. "Conference"). The hypertext command to be executed now is: 'TEXTBOXLOAD TESTR,zzzz,Conference'
- Winisis loads TESTR.PFT, substituting every zzzz sequence by the string "Conference". The resulting format is:

```
Ref(l('Conference'),v24)
```

Winisis will then look up for Conference and return the field 24 of the retrieved record.

k. VIEW database-file-name[,mfn,format-name]

This command is similar to BROWSE, except that the record is displayed in a textbox window rather than a data base window. The main difference is that the window will not contain a toolbar or a status bar and the user will not be able to use any menu function (such as print, search, data entry) on the data base opened with VIEW. For example:

```
link(('Show CDS'),'VIEW cds,15')
```

Note that there is a fundamental difference between TEXTBOX, on the one hand, and BROWSE and VIEW, on the other. The format specified in a TEXTBOX command is always applied to the current record of the data base from which it was activated, while the format specified (or implied) in BROWSE or VIEW applies to the data base opened by these commands.

l. Identifying TEXTBOX and VIEW windows

Each time you click on a TEXTBOX or VIEW hypertext link, CDS/ISIS opens a new window to display the corresponding data. However, having too many windows on the screen may be confusing. You may avoid this by assigning a number to each TEXTBOX and/or VIEW window, so that any further reference to the same number will re-use the corresponding window rather than opening a new one.

The window number (which can be from 1 to 20) must be placed in square brackets immediately after the TEXTBOX or VIEW command, as in the following example:

```
link(('Show record'),'TEXTBOX[2] ',@cds1)
link(('Show record'),'TEXTBOX[1]CHILDLOAD cds1')
link(('Show record'),'TEXTBOX[3]LOAD cds1'
link(('Show CDS'),'VIEW[1] cds,15')
```

In all the examples above, the indicated window will pop up the first time the hypertext link is activated, but any subsequent activation will simply replace the data in that same window.
Note that a numbered TEXTBOX or VIEW window may be shared by two or more data bases.

m. Polymorphic formats

Both the FORMAT command and the LOAD keyword of the TEXTBOX command implement the following optional parameters:

`FORMAT format-name[,`***`old-text`***`,`***`new-text`***`]`
TEXTBOXLOAD format-name[,***old-text***,***new-text***]
which allow you to modify the loaded format *before* its execution: CDS/ISIS will replace each occurrence of the string ***old text*** by ***new text***. This feature allows you to build formats that will act differently depending on the value of ***new text***. For example (assuming that the format cds1 contains the string "xxx") in the following commands:

```
'FORMAT cds1,xxx,',v24
'TEXBOXLOAD cds1,xxx,',v24
```

CDS/ISIS will replace all occurrences of 'xxx' by the content of the field v24 before executing the format ‘cds1’.
Note that ***old text*** must not itself contain a comma, as the format parser will consider the first comma encountered as the separator between ***old-text*** and ***new text***.

n. PRINTTHIS and PRINTSEARCH commands

PRINTSEARCH prints current query results (records) in reversed order through a print dialog option. In case this command is called out of context, i.e. there is no active query, only the current record will be printed.
PRINTTHIS prints the current page, including the hypertext command itself.

```
… link((‘print’),’PRINTSEARCH’)…
```

o. RESETSEARCH (prefix)

Disable the “Display Search Results” mode, as the menu option “Browse”-“Data base” does. This forces Winisis to stop displaying search results. For instance, when search terms highlighting is active, you may use RESETSEARCH to stop highlighting words in order to present a new search interface.
Syntax and parameter description:
Note: RESETSEARCH is a prefix and, at the moment, only applies to command FORMAT.

For example:
```
Link((‘Change’),’RESETSEARCH FORMAT cds1’)
```

p. MESSAGE command

Sends a message to Winisis. Generally it allows opening a data base or a particular dialog box belonging to it. Syntax and parameter description:

```
MESSAGE message,dbasename,formatname
```

message:	the message to be sent.
dbasename:	the dbase that should receive the message.
formatname:	the format to be shown by dbasename.

Every action will open the specified dbase if necessary. Data bases are identified by their name.
Available messages are:
OPENSEARCH	Open the Expert Search.
OPENEASYSEARCH	Open the Guided Search Window
LANGUAGE-XX	Change the actual working language of Winisis. XX should be an existing language code!

q. Example : Hypertext links to another database

This example is to work with two databases, joined by a common field, say v14. After opening the first database, and clicking on the hypertext link, a new window should open in the second database, containing the value of v14 from first database.

Try the following PFT format:

`s1:=v14,` {field from first database} `e1:=LR ->Second_dbase((s1),1,1),` {MFN of first posting of text variable s1}

Hypertext commands are strings, hence write:

```
link (('Second_dbase'),'BROWSE Second_dbase,',f(e1,1,0))
```

The resulting string will be (let's say `e1=50`):

```
BROWSE Second_dbase,50
```

You must close the second window before opening of another record in second database. If you don't close the second window and open another record in first database and then click on the hypertext link, the previous second window is displayed once more.

Alternatively you can use command VIEW. The difference, is that VIEW windows do not display the toolbar for browsing the database. If you just want to show one record at once, when clicking on the hyperlink, you may use command VIEW as follows:

```
link(('click_me'),'VIEW[2] Second_dbase,',f(e1,1,0))
```

Note that you may easily implement a sort of browsing, by implementing hypertext buttons for going to the previous or to the next record. Using ISIS Pascal you can even implement a Search browsing. The following implements a button for going to the next record (up to 100):

```
if mfn<100 then link(('[next]'),'GOTO ',f(mfn+1,1,0)), else ,'[next]',
fi/
```

9. The Field Select Table (FST)

A Field Select Table (FST) defines criteria for extracting one or more *elements* from a master file record. Depending on the context in which an FST is being used, these elements may then be used to create inverted file entries for the record from which they were extracted, for sorting records in the desired sequence before producing a printed report, or to reformat records during an import or export operation.
An element can be generally defined as a fragment of a record resulting from a particular process. Although in many cases *elements* will be actual *data elements,* i.e. a field or a subfield, in other cases they may be words, phrases, or any other piece of data which has a particular meaning to an application.

FSTs are created or modified using the FST editor under Edit > Field Select Table. A sample FST is:

Field ID	Technique	Data extraction format
24	4	`mhl,v24`
69	2	`v69`
70	0	`mhl,v70|%|`
26	0	`"PLACE=",v26^a`
26	0	`"PUBL=",v26^b`
71	0	`(|CORP=|v71/)`
44	0	`(|VOL:|v44^v/,|TIT:|v44^z/)`

An FST consists of one or more lines each defining three parameters:

1. a **field identifier** (column labelled ID);
2. an **indexing technique** (column labelled IT); and
3. a **data extraction format** coded using the CDS/ISIS formatting language.

Whenever CDS/ISIS is requested to extract elements using an FST, it will read the relevant master file records and carry out, for each record and for each FST entry, the following process:

1. execute the format to extract from the record the corresponding data;
2. apply the specified indexing technique to the data produced by the format; and
3. assign to each element thus produced the specified field identifier.

The process described above is strictly *mechanical* and is performed exactly as described. There is no transmission of *knowledge* between one step and the next, only of *data*, although all steps *co-operate* in achieving the desired result. For example, the fact that a particular field was extracted during step 1 is *not* known to step 2: step 1 uses the full power of the formatting language to produce a string of characters and pass it on to the step 2. This step operates on this character string according to the specified indexing technique. Indexing techniques are defined as processes on *character strings,* not on records or fields. It is because of this generalized design that FSTs may be used for such different purposes as defining the contents of the Inverted file or specifying the sorting requirements of a printed listing, which might appear, at first sight, totally unrelated.
In the most general terms, you may think of an FST as a device able to produce elements of data required to perform a certain task.

A. FST parameters

The three parameters of an FST line are described below in the order they are processed (when editing an FST with the line editor, they are entered in the reverse order).

1. Data extraction format

This is coded using the CDS/ISIS formatting language described under "The Formatting Language". Because the data produced by this format is not meant to be displayed, but further processed, CDS/ISIS does not restrict the line width to any particular value and, consequently, it will never split data between lines. The concept of *lines*, however, may be relevant to a particular indexing technique applied to the output produced by the format. In this case CDS/ISIS will guarantee that lines will *only* be created in response to explicit new line commands you specify in the format.

Because of this, certain formatting commands such as the C, the indentation or the escape sequence commands would normally be irrelevant in a data extraction format and may, in some cases, produce unexpected results. They should therefore be avoided, unless they are necessary to achieve the intended result.

On the other hand, the mode (see "Mode command") selected to output certain fields may be instrumental to the correct functioning of a particular indexing technique: certain techniques *require* in fact a specific mode (this is indicated under each indexing technique discussed below). It is your responsibility to insert the appropriate mode command(s) in the data extraction format, if necessary.

Also note that requesting upper case translation, may adversely affect other further processes applied to the data produced by the FST. As a general rule you should *not* request upper case translation (use modes **mpl, mhl or mdl** as applicable, rather than **mpu, mhu or mdu**), unless you are sure it is needed and will not have any side effects. CDS/ISIS will automatically perform upper case translation whenever needed. For example, all elements generated by the Inverted file FST will be translated to upper case before they are stored in the dictionary, even when the FST produces them in lower case.

2. Indexing Techniques

An indexing technique specifies a particular processing to be performed on the data produced by the format in order to identify the specific *elements* to be created. There are eight indexing techniques which you may use. They are given a numeric code from 0 to 4 as explained below.

a. Indexing technique 0

Build an element from each *line* extracted by the format. This technique is normally used to index whole fields or subfields. Note, however, that CDS/ISIS will build elements from *lines*, not from fields. This is because CDS/ISIS looks upon the output of the format as a string of characters where *fields* are no longer identifiable. It is therefore your responsibility to produce the correct data through the format, especially when you are indexing repeatable fields and/or more than one field. In other words, when using this technique, your data extraction format should output one line for each element to be indexed.

b. Indexing technique 1

Build an element from each *subfield* or *line* extracted by the format. As CDS/ISIS will search the output of the format for subfield delimiter codes, for this technique to work correctly your format must specify proof mode (or no mode at all, as this is the default mode), because it is the only mode preserving the subfield delimiter codes on output (remember that heading and data mode *replace* subfield delimiters by punctuation marks). Note that indexing technique 1 is in fact a shortcut to using indexing technique 0. For example: Record content: `^aParis^bUnesco^c1965`

FST	Format output	Elements produced
`1 1 mpl,v26`	`^aParis^bUnesco^cl965`	Paris Unesco 1965
`1 0 mhl,v26^a/v26^b/v26^c`	Paris Unesco 1965	Paris Unesco 1965
`1 1 mdl,v26`	Paris, Unesco, 1965	Paris, Unesco, 1965

c. Indexing technique 2

Builds an element from each term or phrase enclosed in triangular brackets **(<...>).** Any text outside brackets is not indexed. Note that this technique requires proof mode, because the other modes delete the brackets.

The advantages of using triangular brackets over using slashes (Indexing technique 3), are discussed under "Search term delimiters”.

A field containing “`Mission report describing a <university course> in <documentation training> at an East African <library school>`” will produce the following elements when indexed with this technique:

```
university course
documentation training
library school
```

d. Indexing technique 3

Does the same processing as indexing technique 2 except that terms or phrases are enclosed in slashes (/../).

For example the following text:

```
Mission report describing a /university course/ in /documentation
training/ at an East African /library school/
```

will produce the following elements when indexed with this technique:

```
university course
documentation training
library school
```

e. Indexing technique 4

Build an element from each *word* in the text extracted by the format. A word is any sequence of contiguous alphabetic 2 characters[7].

When you use this indexing technique, you may prevent certain non-significant words from being indexed by defining them in a special file called the **Stopword** file (see under “Creating a stopword file” for details on how to build a stopword file).

Note: when this technique is used to index an entire field containing subfield delimiters, you must specify heading or data mode (**mhl** or **mdl**) in the corresponding data extraction format so that subfield delimiter replacement will take place before indexing, otherwise alphabetic subfield delimiter codes will be considered part of a word. It is also advisable to use heading or data mode if the field being indexed contains filing information, so that only the display form of the field is indexed and any data required for sorting the field is ignored (see under “Filing information”).

f. Indexing techniques 5 to 8

The following 4 indexing techniques will allow you to specify a prefix for search terms extracted with indexing techniques 1, 2, 3 and 4. These techniques are numbered 5, 6, 7 and 8 respectively. The prefix is specified in the data extraction format as an unconditional literal as follows:

```
'dp...pd', [format]
```

[7] The definition of alphabetic characters may be customized at each user installation through the system table ISISAC.TAB (see under “Alphabetic characters table (ISISAC.TAB)”.

where
‘d’ is a delimiter of your choice (which does not occur in the prefix itself)
p...p is the actual prefix

For example: `1 8 '/TI=/',v24`

This will index each word of field 24 and prefix each term with TI=.

3. Field identifier

The field identifier is a number (in the range 1-32767) which is assigned to each element created during the indexing step. The meaning of the field identifier depends on the purpose the FST is being used for, as explained below.

Inverted file FST: the field identifier is the qualifier to be used during searching (see under “Operand qualifier”);

Sorting FST: the field identifier is the field tag to be used in a user-supplied heading format (see “Heading format”);

Reformatting FST: the field identifier is the ISO tag to be assigned to an exported field (see under “Reformatting FST”), or the CDS/ISIS tag to be assigned to an imported field (see under “Reformatting FST”).

You may find additional information on FSTs used for a specific purpose under “System sort worksheet”, “Export worksheet”, and “Import worksheet”.

B. *Inverted file FST*

As noted earlier one FST for each data base defines the contents of the corresponding inverted file. The elements built by this FST, once stored in the inverted file, constitute the dictionary of searchable terms for the data base.

The dictionary alone, however, is not sufficient to provide a complete retrieval mechanism, as each term must be linked to all the data base records in which it occurs. Thus to each searchable term in the dictionary, CDS/ISIS associates a list of postings to provide this link. Each term has as many postings as the number of times it occurs in the data base.

Furthermore, in order to support some advanced search language facilities, such as proximity search operators, each posting contains not only the MFN of the record, but also additional information concerning the location of the term in the record from which it was extracted.

A posting has four components:

The **MFN** of the record containing the term; CDS/ISIS assigns this component automatically.

The **field identifier** to be used during searching when an operand qualifier is specified (see “Operand qualifier”). You assign this component through the FST. Note that you may assign the same field identifier to different fields and make them therefore appear as one field when searching.

The **occurrence number**, required to implement field level search operators (**(F)** and proximity operators) on repeatable fields. If the use of these operators is a design requirement of the data base (which will normally be the case when using indexing technique 4), then you must assign the occurrence number, through the format, by outputting a percent sign (%)[8] between occurrences of a repeatable field

[8] Note that the actual character to separate the occurrences of a repeatable field is an installation-dependent parameter (see “Parameter 8: Repeatable fields separator”). If this has been re-defined, you must use the character assigned for this purpose rather than the % sign

(which you may easily do by using a repeatable suffix-literal, e.g. `v10|%|`). CDS/ISIS resets the occurrence number to 1 at each FST line processed, and increases it by 1 each time it encounters a % sign in the text produced by the format.

The term **sequence number**, required to implement proximity search operators. This is assigned by CDS/ISIS as follows: it is reset to 1 at each FST line processed and each time the occurrence number is changed; it is increased by 1 for each element created by the specified indexing technique. Assume, for example, that a data base has a summary in repeatable field 68, containing one paragraph per occurrence, which must be indexed using indexing technique 4. By specifying the data extraction format **`mdl,v68|%|,`** the words of the summary will be assigned a sequence number starting from 1 at each paragraph (whereas if the data extraction format was **`mdl,v68`**, the words would be numbered sequentially for the whole summary, e.g. the first word of the second paragraph will have a sequence number which is one higher than the last word of the first paragraph).

In some cases, certain dictionary terms generated by the Inverted file FST may prove to be meaningless or ambiguous for an efficient retrieval. For example, the term **IT** could be either a pronoun (in which case it would be useless in retrieval), or the country code of **Italy**. This would also be true for numeric terms such as **34**, whose meaning may be obscure when isolated from its context. If the field producing such terms is indexed with technique 0, and 5 to 8, you may add meaning by using a prefix-literal to provide a label which will then identify the particular usage of the term. For example, when indexing a field containing a country code, e.g. field 10, you could use the following format: **“CC = ”v10** (rather than **v10**), so that the country code **IT** will be inverted as **CC=IT** (rather than just **IT**). Likewise, if field 20 contained a project number, you could use the format **“PROJECT ”,v20**, so that you may later search for **PROJECT 34** (rather than just **34**).

1. Testing and modifying the Inverted File FST

The inverted file (ie Dictionary) is pivotal to the efficient retrieval of information from a database. To ensure that the required access points are in the inverted file, it is useful to test the FST data extraction formats. You should choose a representative sample of records in the database, and display them one at a time (see p.40, The Database window), applying any format except *ALL*. Click the "Split" button to bring up the format editor. Type the data extraction format(s) to be tested into the editor window, click the button to execute the format(s), and view the output in the top window. In particular, check the following:

- If you apply indexing technique 0 to a repeatable field, is each occurrence on a separate line?
- If you apply indexing techniques 4 or 8 to a repeatable field, is there a repeatable field separator (usually, % character) between each occurrence of the field? (if not, the field level and proximity search operators will not work correctly).
- If you apply indexing techniques 2,3,6 or 7, can you actually see the enclosing delimiters around each term to be indexed? (if not, make sure that your format uses proof mode).
- If you apply indexing techniques 1 or 5, can you actually see the subfield delimiters on screen? (if not, make sure you use proof mode, as above).
- If you apply indexing techniques 4 or 8 to a field containing subfield delimiters(^) or filing information (<,>), does the output show these delimiters? (if yes, make sure you use heading or data mode, so that they are replaced before indexing).

If there appears to be something wrong, edit the format(s), execute, and examine the output, as above.

Although you can easily modify the FST in a text editor, be aware that a full inverted file generation may be needed, to remove inconsistencies and dead pointers. In general, changing the indexing technique, data extraction format, or field identifier will require a full inverted file rebuild. Adding or deleting FST lines will also require a full inverted file rebuild. However, if you have added an FST line corresponding to a field which did not previously exist, then only an inverted file update is needed.

The next tables show the processing of an FST and the full posting assigned to each element before it is stored on the Inverted file.

Elements produced		
FST line	**Format output**	**ID element**
24 4 mhl ,v24	<An> Electric hygrometer apparatus for measuring water-vapour loss from plants in the field	24 An 24 Electric 24 hygrometer 24 apparatus 24 for 24 measuring 24 water 24 vapour 24 loss 24 from 24 plants 24 in 24 the 24 field
69 2 v69	Paper on: <hygrometers><plant transpiration><moisture><water balance>	69 hygrometers 69 plant transpiration 69 moisture 69 water balance
70 0 mhl,v70|%|	Grieve, B.J.%Went, F.W.%	70 Grieve, B.J. 70 Went, F.W.
26 0 "PLACE="V26^a	PLACE=Paris	26 PLACE=Paris
26 0 "PUBL="V26^b	PUBL=Unesco	26 PUBL=Unesco

MFN	**ID**	**OCC**	**SEQ**	**Element**
4	24	1	1	An
4	24	1	2	Electric
4	24	1	3	hygrometer
4	24	1	4	apparatus
4	24	1	5	for
4	24	1	6	measuring
4	24	1	7	water
4	24	1	8	vapour
4	24	1	9	loss
4	24	1	10	from
4	24	1	11	plants
4	24	1	12	in
4	24	1	13	the
4	24	1	14	field
4	69	1	1	hygrometers
4	69	1	2	plant transpiration
4	69	1	3	moisture
4	69	1	4	water balance
4	70	1	1	Grieve, B.J.
4	70	2	1	Went, F.W.
4	26	1	1	PLACE=Paris
4	26	1	1	PUBL=Unesco

The table below shows the elements and corresponding postings that would be generated if a **stopword** file was used. Note, in this example, that, although stopwords are not output, they are nevertheless accounted for in the term sequence number.

MFN	ID	OCC	SEQ	Element
4	24	1	2	Electric
4	24	1	3	hygrometer
4	24	1	4	apparatus
4	24	1	6	measuring
4	24	1	7	water
4	24	1	8	vapour
4	24	1	9	loss
4	24	1	11	plants
4	24	1	14	field
4	69	1	1	hygrometers
4	69	1	2	plant transpiration
4	69	1	3	moisture
4	69	1	4	water balance
4	70	1	1	Grieve, B.J.
4	70	2	1	Went, F.W.
4	26	1	1	PLACE=Paris
4	26	1	1	PUBL=Unesco

Stopword file

AN
FOR
FROM
IN
THE

10. Adapting CDS/ISIS to local requirements

A. Creating a new language version

In the Windows version of CDS/ISIS both menus and system messages are stored in CDS/ISIS data bases (unlike the MS-DOS version, where menus were stored in special menu worksheets and messages in a special data base). To create a new language version of the Windows version of CDS/ISIS you must create the corresponding menu and message data bases for the new target language.

Menus are stored in menu data bases located in the menus directory, indicated by parameter 2 of syspar.par (by default **\winisis\menu**). The names of menu data bases are made up as follow:

Mn*xxyy*

where:

xx is the language code (i.e. **EN** for English). You may define new codes, such as CZ for Czech, or PL for Polish, the only restriction being that you may not have two different languages coded with the same acronym.

yy is the profile code. The default code is **DF**, which includes all the CDS/ISIS menu commands. You may, in addition, define your own menu profile, e.g. BS for a menu that, for example, allows you the user to only access certain functions (see below under *Creating a new menu profile*).

The name of the initial menu data base to be used by CDS/ISIS is defined in parameters 101 and 102 of syspar.par.

Messages, including labels used in the various CDS/ISIS windows and dialog boxes, are stored in message data bases located in the message directory, indicated by parameter 3 of syspar.par (by default **\winisis\msg**). The names of message data bases are made up as follows:

Ms*xx*

where:

xx is the language code (i.e. **EN** for English). You may define new codes, such as CZ for Czech, or PL for Polish, the only restriction being that you may not have two different languages coded with the same acronym. The language code used for a message data base *must* be the same as the one used in the corresponding menu data base.

1. Creating a new menu data base

This example explains how to create a Spanish version of the DF (default) menu. Perform the following steps:

1. Choose the new acronym for your language (for example, SP for Spanish).
2. Open a DOS prompt in Windows.
3. Move to your Winisis directory. Normally it should be C:\WINISIS.
4. Move to the menu directory. (C:\WINISIS\MENU)
5. Execute the following commands:

```
xcopy mnendf*.* mnspdf*.*
copy ?mnspd.fmt ?mnspd.fmt
```

This creates a copy of the default English menu into the Spanish files. In particular, the first statement will copy the menu database, while the second will copy the data entry worksheets for that database.

6. Once you have created a copy of the default menu, you may close the DOS window and execute Winisis. Choose ***Open*** from the **Data base** menu and select your menus directory (by default \WINISIS\MENU) and choose the newly created data base MNSPDF.MST.

7. Open the Data Entry window and, browsing the database a record at a time, translate each command to the new language (Spanish).

Note. Each menu command is followed by a number or a keyword enclosed in square brackets (e.g. [64]): they represent the meaning of each command to CDS/ISIS. *You must not delete or modify any of these numbers. If you do, CDS/ISIS will not work properly when this language version is used.*

8. Close the database.

9. Before you can test your new menu, you must also create a new language version of the messages corresponding to the language code you used (in this example SP) as explained below.

2. Creating a new message data base

This example shows how to implement a new language message data base, for example Spanish.

1. Choose the new acronym for your language (for example, SP for Spanish). Note that you must use the same language acronym for both the menu and the corresponding message data base.

2. Open a DOS prompt in Windows.

3. Move to your Winisis directory. Normally it should be C:\WINISIS.

4. Move to the messages directory. (C:\WINISIS\MSG)

5. Execute the following:

```
xcopy msen.* mssp.*
copy ?msen.fmt ?mssp.fmt
```

This creates a copy of the English message data base in a new database (MSSP.*).

6. Close the DOS window and execute Winisis. Choose the ***Open*** command from the **Data base** menu and select the message directory (by default C:\WINISIS\MSG) and choose the newly created data base MSSP.MST.

7. Open the Data Entry window and, browsing the database one record at a time, translate each message in the new language. The following table will help you to understand the meaning of each message in the MST:

MFN Range	Message category
1-10	Standard buttons (OK, CANCEL, etc.) Note that the first field of the first record contains the language name that will be displayed by Winisis when showing the list of available languages.
11-12	Database and record status (Data base window).
12-16	Configuration, import, export and Inverted file maintenance Dialog boxes.
21-22	Search windows.
31	Open Dialog box
41-42	Error messages.
51-52	Data Entry window
61-70	Print Dialog box
71-73	Tool bars help messages.

You may refer to the languages.doc file for more information about the newest messages added to Winisis.

In order to set the new language as the default language (i.e. the one which CDS/ISIS will automatically select at startup) you must set parameter 101 of syspar.par equal to the corresponding language code, for example:

```
; Default language
101=SP
```

3. Creating a new menu profile

By using *menu profiles* you may restrict the set of CDS/ISIS commands which a given user may have access to. A menu profile is in fact a particular menu data base which only contains a subset of CDS/ISIS commands. Normally a menu profile is selected through parameter 102 of syspar.par. However, by means of parameter 0, you may assign a different profile to each user (or category of users), by redefining parameter 102 for each secondary syspar.par. By default CDS/ISIS will use the **DF** profile (data base **MNxxDF**), which allows access to all CDS/ISIS functions.

To create a new menu profile for an existing language version, perform the following steps:

1. Choose the new acronym for your profile (for example, **BS**).
2. Open a DOS prompt in Windows.
3. Move to your Winisis directory (by default C:\WINISIS).
4. Move to the menu directory. (C:\WINISIS\MENU)
5. Execute the following:

   xcopy mnendf*.* mn*xx*bs*.*
   copy ?mnend.fmt ?mn*xx*b.fmt

This creates a copy of the ***xx*** language menu data base. In particular, the first statement will copy the database, while the second will copy the data entry worksheets for that database.

Note: data entry worksheet names (.FMT files), are composed by a letter indicating the page number and by the first 5 characters of the database name (for example AMNEND.FMT). So, if you have more than one profile beginning with the letter "B", say MNENBS.MST and MNENBR.MST, you do not need to duplicate the entry worksheets, because in both cases the worksheet name is AMNENB.FMT. This is not a problem and does not cause any conflict between the two profiles.

6. Once you have created a copy of the default profile, you may close the DOS window and execute CDS/ISIS. Choose the ***Open*** command from the **Data base** menu, and select your menu directory (by default C:\WINISIS\MENU), then choose the newly created MST file, MNENBS.MST.
7. Open the Data Entry window and, browsing the database one record at a time, delete all the commands you do not want to be available in the new profile.

Note: to avoid the user to select the default menu, you should delete the ***Change language*** command in the **Configure** menu. You should also change the profile name, by editing the first field of the first record in the database, that normally says: "Full Version".

8. Close the database.
9. You may now test your new profile, by choosing the new profile from the ***Change profile*** command in the **Configure** menu. If something is wrong and you cannot return to the full version in order to make additional changes, close and reload CDS/ISIS, after removing parameter 102 of syspar.par, if any.
10. To set the new profile as the default one, change parameter 102 of syspar.par.

4. Renaming a database

First backup the existing database. Then copy all the database files into a new folder with the new name, in your DATA folder. Rename the files (using a maximum of 6 characters for the new database name).

1. Open the new FDT file in a text editor. This file tells the system which worksheets (*.FMT), formats (*.PFT) and field select tables (*.FST) the database uses, on the first three lines of the FDT file, eg:

```
W:ILOAN
F:ILOAN ILAP  ILREP
S:ILOAN EXPO
***
```

Change the file names on these three lines to the new name(s), allowing exactly 6 characters, with spaces if needed, for each name. The first name on each line is always the name of the database.

2. The program monitors the dates of the various files generated when creating a database. The FDT file (the first file created) must have an earlier date than the FMT, PFT and FST files, otherwise you will get an error message. The solution is simply to open each file in a text editor, make a sham modification (eg inserting a space and then deleting it), and then close and save the file, which will now have a later date.
3. Create a new **dbn.par** file in your DATA folder. Erase the database (in the Utilities menu) if a new, empty database is required, and do a full Inverted File rebuild.

B. Conversion tables

In order to maintain compatibility with the MS-DOS version, the Windows version of CDS/ISIS maintains data bases (on disk) using the MS-DOS character set (known as the *OEM character set*). The data is then converted to the Windows character set (known as the *ANSI character set*) whenever it is displayed on the screen and converted back to MS-DOS whenever you update a record. In this manner the same physical data base can be operated using both the MS-DOS and the Windows version and can be exchanged with other CDS/ISIS users independently of the version used. Depending on the language of the data base and the corresponding MS-DOS code page used, you may have to provide conversion tables if the default built-in tables are not working properly[9]. All conversion tables are text files containing 256 integers (except ISISAC.TAB), corresponding to the character codes 0-255. Character conversion is performed by replacing a given text character by the corresponding code in the table. For example, if the 129th integer in the table is 199, then CDS/ISIS will replace each occurrence of character code 128 with character code 199. The conversion tables used by CDS/ISIS are described below.

1. OEM to ANSI conversion table

This table is used to convert the data from the DOS character set used in the data base to the Windows character set. By default CDS/ISIS will use a built-in table. If you provide this table, you must enter its path and file name in parameter 106 of syspar.par. We recommend you store the table in the menu path (parameter 2) and call it OEMANSI.TAB. Note that in this case you must also provide the Windows to DOS conversion table described below.

2. ANSI to OEM conversion table

This table is used to convert the Windows character set to the DOS character set before updating a record. By default CDS/ISIS will use a built-in table. If you provide this table, you must enter its path and file name in parameter 107 of syspar.par. We recommend you store the table in the menu path (parameter 2) and call it ANSIOEM.TAB. Note that in this case you must also provide the OEM to ANSI conversion table described above.

3. Upper case conversion table (ISISUC.TAB)

This table is used to convert data base text (i.e. as stored in the data base using the MS-DOS character set) to upper case. It must be stored in the menu path (parameter 2 of syspar.par). The standard table supplied by UNESCO is given below:

```
000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015
016 017 018 019 020 021 022 023 024 025 026 027 028 028 030 031
032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047
048 049 050 051 052 053 054 055 056 057 058 059 060 061 062 063
064 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 091 092 093 094 095
096 065 066 067 068 069 070 071 072 073 074 075 076 077 078 079
080 081 082 083 084 085 086 087 088 089 090 123 124 125 126 127
```

[9] CDS/ISIS uses as default the standard Windows ANSI-to-OEM mappings, which are defined by the keyboard driver, where this function is implemented. Some keyboard drivers may have different mappings than others, depending on the machine environment, and some keyboard driver support loading different OEM character sets; for example, the standard U.S. keyboard driver for an IBM keyboard supports loadable code pages, with the default being code page 437 and the most common alternative being code page 850. (The Windows character set is sometimes referred to as code page 1007.)

```
067 085 069 065 065 065 065 067 069 069 069 073 073 073 065 065
069 069 069 079 079 079 085 085 089 079 085 155 156 157 158 159
065 073 079 085 078 078 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255
```

4. Alphabetic characters table (ISISAC.TAB)

This table defines the character codes of all *alphabetic* characters. It is used each time CDS/ISIS needs to know whether a given character is alphabetic (e.g. when performing word indexing using indexing technique 4, or validating alphabetic fields). A given text character whose code is stored in this table (using the MS-DOS character set) will be considered an alphabetic character. The standard table supplied by UNESCO is given below:

```
065 066 067 068 069 070 071 072 073 074 075 076 077 078 079 080
081 082 083 084 085 086 087 088 089 090 097 098 099 100 101 102
103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118
119 120 121 122 128 129 130 131 132 133 134 135 136 137 138 139
140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 160
161 162 163 164 165
```

11. CDS/ISIS Pascal

CDS/ISIS Pascal programs are currently implemented only as format exits. The programs are fully compatible at .pcd level with the DOS version of CDS/ISIS. However, certain restrictions in the MS-DOS version, in particular regarding maximum string size and number of strings, no longer exist. A string may now be up to 32K, including the input and output strings of a format exit.

Note that, at present, only the following ISIS library routines are available.

ASSIGN
CHR
COPYSTR
DBN
DATESTAMP
EDIT
ENCINT
ENCREAL
EXEC
FIELD
FIELDN
FILEXIST
FIND
FLDTAG
GETMFN
LANG
MAXMFN
MAXSET
NFIELDS
NOCC
NPOSTS
NXTPOS
NXTPOST
NXTTERM
ORD
PATH
POSITION
POSTING
READ(ln)
RECALL
RECORD
SEARCH
SETPOS
SIZE
SUBSTR
SYSTEM
UC
VAL
WRITE(LN)

Also note the following implementation differences between the MS-DOS and the Windows version of the functions listed below.

i. ASSIGN

The **/k** switch (to prevent closing a file between calls) is not supported.

ii. READ(ln) and WRITE(ln) procedures

These functions are implemented only for files (i.e. Read(**inp**,...) and Write(**out**,...)). Input from keyboard or output to the screen are not supported. This implies that you must always explicitly open the file to be read or written by using the **ASSIGN** procedure.

iii. EDIT

Function EDIT is implemented as follows: it serves to edit at run time any string passed to the function. When calling EDIT, a mini-dialog box appears on the screen to make it possible to modify the given value. Click Ok to validate the entry.

Syntax: EDIT(var: s: string, n1,n2,n3,n4,n5: real, title: string)

Parameters list:

S the string to be edited (after EDIT s will contain the resulting edited string). S may be initially empty.

N1,n2,n3,n4: not used

N5 May assume two values: Single (**1**) or Multiple (**2**) lines edit

Title is a **%** delimited string containing: the title of the dialog window%a first line of comment%a second line of comment. Example:
Edit Search Expression%You can edit the search expression%Click Ok to execute the search.

The dialog title is: Edit Search Expression
The first comment line is: You can edit the search ....
The second comment is: Click Ok to execute the search

iv. EXEC procedure

1. All programs called through EXEC must be of the same type as the original caller (e.g. if program A is a format exit and it calls program B then B must also be a format exit);
2. If the calling program (A) is a format exit, then the called program (B) receives as input string (s1), the current value of the output string (s2) of A at the time the EXEC function is executed. Program B returns a value in the output string (s2). This allows you to pass an input parameter to, and to receive output from the executed program. For example:

```
Program A(s1: string; lw, occ: real; s2: string) [format];
begin
s2:='xxx';
exec('B');
{ the value of s2 at this point will be 'xxxyyyzzz' }
s2:=s2|'www';
end.

Program B(s1: string; lw, occ: real; s2: string) [format];
begin
{ on entry: s1 will be 'xxx' }
s2:=s1|'yyy';
exec('C');
{ the value of s2 at this point will be 'xxxyyy' }
end.

Program C(s1: string; lw, occ: real; s2: string) [format];
begin
{ on entry: s1 will be 'xxxyyy' }
s2:=s1|'zzz';
{ the value of s2 at this point will be 'xxxyyyzzz' }
end.
```

On exit from A, the value of s2 will be 'xxxyyyzzzwww'.

vi. LANG

Return a two character string containing the current selected language code.

vii. PATH function

Not implemented yet.

Until such time as the **dbn.par** facility will be implemented, the function PATH('DBN',i) returns the path of the current Master file for all values of i.

viii. SYSTEM procedure

This procedure may execute both DOS and Windows programs. If you are executing a DOS program and you want it to execute in a window (rather than in full screen mode), you must also create a corresponding PIF file.

The program called by the SYSTEM procedure executes asynchronously and control returns immediately to your program. This feature may be used, for example, to play a sound file (e.g. a description of the record) while the record is displayed. Assume for example that field 1 contains the name of a .wav file associated with the record. Then the following format, and the associated format exit, will play this file while displaying the record:

```
Format: ............ &play() ..........


Program play(s1: string; lw, occ: real; s2: string);
var wav: string
begin
wav:=field(fieldn(1,1));
if (size(wav)>0) then system('mplayer /play /close '|wav|'.wav');
end.
```

App. A - Windows editing keys

The following table provides a summary of the standard Windows editing keys (the + sign indicates that you must simultaneously press the indicated keys):

CTRL+INS or **CTRL+C**	copy the selected text into the clipboard
SHIFT+DEL or **CTRL+X**	move the selected text into the clipboard and delete it from the screen
SHIFT+INS or **CTRL+V**	insert (paste) the clipboard content to the current cursor position
CTRL+V	paste from the clipboard
ALT+Backspace or **CTRL+Z**	*undo* the last editing operation
Esc	(in data entry) reload the field content from the disk.
← (Cursor left)	move the cursor one position to the left
CTRL+← (Cursor left)	move the cursor one word to the left
→ (Cursor right)	move the cursor one position to the right
CTRL+→ (Cursor right)	move the cursor one word to the right
↑ (Cursor up)	move the cursor to the same position in the preceding line (if within the field) or to the beginning of the field
↓ (Cursor down)	move the cursor to the same position of the following line (if within the field) or to the end of the field
↖ (Cursor Home)	move the cursor to the beginning of the current line
CTRL+↖ (Cursor Home)	move the cursor to the beginning of the field
Cursor End	move the cursor to the end of the current line
CTRL+Cursor End	move the cursor to the end of the field
Cursor PgUp	move one page up (if within the field) or to the first line of the field
Cursor PgDn	move one page down (if within the field) or to the last line of the field
Backspace	delete the character to the left of the cursor and move cursor one position to the left
Delete	delete the character at the cursor position (the cursor does not move)
Shift+Cursor key	select some text

Glossary

Access point : see Search term

ANY file : An optional file containing the set of ANY terms defined for a given data base. The ANY file is a text file which may be created with any text editor (e.g. NOTEPAD or WRITE)

ANY term : A collective term representing an arbitrary (but pre-defined) set of search terms. When used in a search expression, an ANY term is automatically translated to the set of search terms it represents linked with the logical OR operator. If, for example, the ANY term ANY BENELUX represents the search terms BELGIUM, NETHERLANDS and LUXEMBOURG, then when ANY BENELUX is used in a search expression it will be translated to (BELGIUM + NETHERLANDS + LUXEMBOURG). Before an ANY term can be used in a search expression, it must be defined in the ANY file

ASCII : (American Standard Code for Information Interchange) A 7-bit binary code used to encode characters on a computer. ASCII codes range from 0 to 127. On many computers, such as the IBM PC, the code has been extended to 8 bits, providing therefore an additional set of 128 codes from 128 to 255. These are not normally part of the ASCII standard and cannot therefore be expected to be the same on all computers

Browsing : the sequential display of Master file records in MFN order, i.e. in the order in which the records have been entered

Control key : Keys on your keyboard which, when pressed, perform a specific pre-defined action

Cut and paste : An editing operation, available in most word processing packages, consisting in moving one piece of text from one place to another. The CDS/ISIS field editor supports the standard Windows cut and pasting operations

Data base : A file of related data collected to satisfy the information requirements of a given user community. Each unit of information stored in a data base consists of discrete *data elements*, each containing a particular characteristic of the entity being described. For example, a bibliographic data base will contain information on books, reports, journal articles, etc. Each unit will, in this case, consist of such data elements as author, title, date of publication, etc. Data elements are stored in *fields*, each of which is assigned a numeric *tag* indicative of its contents

Data element : An elementary piece of information which CDS/ISIS can identify. A data element may be stored in a field or a subfield

Data entry worksheet : An electronic form used for entering data in a data base

Default value : The pre-defined contents of a field

Dialog box : An electronic form used by CDS/ISIS to collect the parameters for a particular operation such as a print run or an export operation

Dialogue language : The language used by CDS/ISIS to interact with the user. CDS/ISIS is designed to support an unlimited number of dialogue languages

Dictionary : The set of search terms for a given data base

Display format : see Format

Display mode : The manner in which fields are displayed. CDS/ISIS may display fields in three different modes: proof, data or heading mode

Export : The conversion of a data base from the form in which it is stored for processing to a form suitable for transmission to other users or systems

FDT : see Field Definition Table

Field : The *container* of a data element

Field Definition table : A table defining the fields of a given data base

Field Select Table : A table defining criteria for extracting one or more elements from a Master file record required for a particular process. Field Select Tables may be used for defining the contents of the

Inverted file, for sorting records before producing a printed report or to reformat records during an import or export operation

Filing information : A special coding inserted in a field defining how it must be sorted

FST : see Field Select Table

Function key : see Control key

Gizmo : A special encoding of accented characters or other special characters used for the exchange of data between the various versions of CDS/ISIS. A gizmo represents a single character and it consists of 3 characters, the first of which is always an @ sign and the next two define the character being represented. By providing gizmo conversion tables at each end, two users are able to transmit and receive correctly characters having different machine codes.

Hit : A record satisfying a set of specific search criteria

Hit file : A file created by CDS/ISIS to sort Master file records

Hit list : The set of records retrieved by a given search expression

Import : The conversion of a data base from the form suitable for transmission to other users or systems to the form required by CDS/ISIS for processing

Indexing : The process of extracting search terms from a Master file record

Inverted file : A logical structure built automatically by CDS/ISIS to enable fast retrieval. It contains the dictionary of search terms and, for each term, a list of references to the Master file records from which the term was extracted. The Inverted file actually consists of six physical files

ISO : International Standards Organization

ISO 2709 : An international standard format for information interchange recommended by ISO

Language : see Dialogue language

Limits : see MFN limits

Link file : A file built by CDS/ISIS during the process of creating the Inverted file

Master file : A logical structure containing the records of a given data base. The Master file actually consists of two physical files

MFN : (Master File Number) the unique number automatically assigned by CDS/ISIS to each record entered in a data base. MFNs are assigned sequentially, starting from 1, and represent the chronological order of entry

MFN limits : A range of MFNs, e.g. 1/50, defining the Master file records to which a certain operation is applied

Mode : see Display mode

Occurrence : One instance of a repeatable field

Paste : see Cut and paste

Pattern : A character by character description of the possible contents of a field. A pattern defines, for each position of the field, the type of characters it may contain (e.g. alphabetic, numeric, etc.)

Print format : see Format

Qualifier : A construct used, during searching, to specify the field or fields in which a given search term should appear

Record (Master file) : The set of fields containing all the data elements of one information unit stored in a data base

Repeatable field : A field which may occur more than once in a record

Save file : A file used to save the results of a search

Search term : Any of the elements that can be used to retrieve a record, e.g. a subject descriptor, a name, a word, a document code, etc. The search terms for the records of a given data base are defined in the inverted file FST

Stopword : A non significant word to be ignored when indexing a field (e.g. articles, prepositions, etc.)

Subfield : One of the data elements contained in a field. A subfield is identified by a subfield delimiter

Subfield delimiter : A 2-character code preceding and identifying a subfield of a field

Tag : A number uniquely identifying a particular field

LIST OF FIGURES

INDEX