ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

DEVELOPMENT OF SOME UTILITIES FOR MICRO CDS-ISIS SOFTWARE

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

BY

MOHAMED RASHID LUGONGO

JUNE 1994

DECLARATION

The thesis is my original work and has not been presented for a degree in any other university.

Mohamed Rashid Lugongo

May 31, 1994

The thesis has been submitted for examination with our approval as university advisors.

Professor A. Neelameghan
Advisor
May 31, 1994

Dr. Gobinda G. Chowdhury
Advisor
May 31, 1994

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF INFORMATION STUDIES FOR AFRICA

DEVELOPMENT OF SOME UTILITIES FOR MICRO CDS-ISIS SOFTWARE

BY

MOHAMED RASHID LUGONGO

Approval by the Board of Examiners:

Getachew Birru — Chairman, School Graduate Committee

A. Neelameghan — Advisor

G. G. Chowdhury — Advisor

T. K. Ravichandra Rao — Examiner

G. Bhattacharyya — Examiner

Taye Tadesse — Examiner

ABSTRACT

The current research examined the need for an online data validation facility, specifically online data entry assistance and online help on format errors for databases created using MicroIsis (Micro CDS-ISIS) software package.

To achieve the specific objectives of this study, two CDS-ISIS Pascal programs (DEA and MOCDS) have been developed using the CDS-ISIS Pascal language.

DEA program is divided into two main parts. The first part is developed to provide online field-by-field guidance to data entry operators on how to select and enter data in different fields in a record of a MicroIsis database, based on a machine-readable data entry guide. For the purpose of testing and demostration of data entry assistance, the ABNCD database guide (accommodating bibliographic records, as well as records of profiles of experts, institutions, projects and information systems), is used. The second part is developed to provide guidance on 'Authority files' for certain data elements, especially for the names of corporate bodies, names of personal authors, project titles, monographic series, serial titles, country codes, language codes, meetings, etc. The program (DEA) provides online guidance on how to enter authority records and provides possibility of marking certain data elements and copying them to similar

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVES OF THE STUDY

The quality of information output and of the services of an information system is contingent upon the quality of input data entered into the system. The software for creation and maintenance of records in a database must not only provide for the different ways in which the output is to be presented to the end-user but should also provide a means of ensuring high quality of input data into the system in order to produce high quality of output information.

The general objective of the present work is to improve the quality of the information services by ensuring some aspects of the quality of input data entered into the system with particular reference to databases built using Unesco's Micro CDS-ISIS or MicroIsis software. The specific objectives are:

- To develop programs using CDS-ISIS Pascal language to assist users of MicroIsis at three levels:
  - To provide field-by-field online guide to data entry into MicroIsis databases;

- To provide an online facility for selecting data from appropriate authority list(s) or vocabulary control tools and add the selected data to specified field(s) in one or more records in a database; and
- To provide an online help facility to handle format errors that may occur while preparing/specifying display formats.

1.2 BACKGROUND AND JUSTIFICATION

1.2.1 Automation of Information Systems in Africa

Developments in microelectronics applications, especially in microcomputers, have had and continue to have significant impacts on information processing. Traditional information processing tasks are being performed much faster, and new methods of handling information are being constantly researched. Emerging information technologies enable the presentation of information in such forms and formats convenient to users and at the same time improving the quality of output. The latter is secured by ensuring the quality of input data.

Until about the beginning of the 1980's the types of computers on the market were mostly mainframes and minicomputers affordable and used mainly by large

organizations with large volumes of transactions and adequate resources. However, in the past decade following the advent and wider availability of microcomputers with high speed data processing capacity, high storage capacity and low cost, application of computers to information processing has gained wider and faster acceptance in most disciplines and even in small institutions. It has encouraged applications in small libraries and information centres even in developing countries.

The use of microcomputers and related technologies in libraries and information systems in developing countries has been promoted and/or supported by international organizations, such as, the specialized agencies of the United Nations and by non-governmental agencies investing in development projects and they have, for example, provided technical and financial assistance to developing countries to:

- build indigenous capability for managing information and promoting its use;
- improve information systems, services and tools for managing and using information relevant to development research and change; and
- encourage cooperation through distributed data processing.

The assistance provided includes the provision of

machines and equipment (e.g. computers, software), manuals, training of personnel, and financial support through various projects relating to the field of information science.

Application of microcomputers to information storage and retrieval systems in specialized libraries and information centres has gained wide and fast acceptance in Africa during the past five years. In part this has been due to the establishment of the Pan African Development Information System (PADIS) by the United Nations Economic Commission for Africa in Addis Ababa with International Development Research Centre (IDRC) support. PADIS has provided training to library and information professionals in the use of microcomputers in information processing and also, in some cases, provided microcomputer facility. PADIS surveys of 1988 and 1990 have confirmed the increasing use of microcomputers and of MicroIsis software in Africa (PADIS 1988; 1990). Use of MicroIsis in Africa has been spreading, thanks to PADIS, supplementing the efforts of the United Nations Educational, Scientific and Cultural Organization (Unesco).

PADIS is currently experimenting on a computer communications network to improve the flow of development information in Africa aimed at contributing towards the efficient and effective utilization of information

technologies through strengthening online linkages between existing information systems and networks in the region. The project is supported by IDRC.

Other international organizations, such as, the International Livestock Centre for Africa (ILCA) have been promoting microcomputer applications and the use of MicroIsis in national agricultural and livestock development institutions in Africa.

The use of MicroIsis has also significantly increased in other parts of the world in the 1990's. There is an estimated twenty thousand or more users of the software (International Classification (1993)).

In the development and use of information storage and retrieval systems, the creation of an information base (data store, database) is a prerequisite. As already mentioned, the quality of information and information products output and therefore of the information services depend very much on the quality of data input. An aspect of the input preparation is to ensure consistency and uniformity in the use of terms to represent/name subjects, countries, languages, objects, corporate bodies, serial titles, etc. The naming and representation of some of these entities in catalogue entries for instance, are guided by widely accepted guidelines (e.g. Anglo-American Cataloguing Rules (AACR2), thesauri, coding schemes, standards, etc). Ensuring consistency and

uniformity in this respect is all the more important if the databases are to be used by a large number and wide range of users, or if the records are to be exchanged among different systems as may be necessitated in networking. Most software packages including MicroIsis do not provide an online facility to select the required data from authority lists, coding schemes, terminology lists, etc. and add it to specified fields in one or more records in a database at the data entry stage. Some packages including MicroIsis do provide facility for online use of thesauri, etc. at the stage of formulating search expressions. MicroIsis provides for linking up for display purpose, data in different records in the databases resident in the same drive through format exit devices such as REF and REF(L) functions and pascal programs. Manually consulting the authority lists, etc. for selecting appropriate standard form of entry and then entering the data onto the worksheet of the record concerned is cumbersome and time consuming. Further, there are greater chances of occurrence of errors in keyboarded data than when the whole process is done online.

1.2.2 Literature review

As already mentioned in section 1.2.1 most software packages including MicroIsis do not provide adequate online facility at the input and data processing stages to guide users. The online facility could be of help in the following ways:

- To provide easy to use facility to users who may only need to make occasional data entry, to do so without having to master the complete system; and
- To enable intermediate/experienced users to explore the full potential of all the program options available.

Most software developers in particular those dealing with database management system (DBMS) software and text retrieval software have been attempting to provide help facilities to their users especially at the search and retrieval stage. Significant activities to tailor various systems using the available facilities in-built into the systems to suit end-user requirement particularly at the search and retrieval phase have been done by different individuals and/or institutions. Several utilities using CDS-ISIS Pascal language of MicroIsis are available to help users in formulating search expressions. Molla Hunegnaw (1993) developed a Search Interface System Assistant (SISA) in an attempt at providing end-user

interface at the search and retrieval stage to assist users of MicroIsis. Srilatha and Aparna (1992) modified the TEXT.PAS program to develop INFRET.PAS that is capable of listing database names and fields of the selected database facilitating selection of database and search fields in free text searching. Neelameghan (1991) used THES.PAS program to enable end-user to navigate from a broad subject area to the specific concepts of interest to them in a specialised field to formulate search expression(s). This has been applied to an information system to support urban planning (Teshome Bula 1993). With version 3.07 of MicroIsis an interface for search and retrieval is now available (Heurisko).

With the view to minimizing typing errors in data entry into databases, the standard MicroIsis has facility for entering default data in one or more fields of a database, either permanently or for temporary use. The more permanent default values that may occur in one or more fields in many of the records of a database, can be entered at the time of preparing the related worksheet. This will obviate the need to enter those default values at the time of data entry and thus minimizing typing errors in data entry in those fields.

MicroIsis also permits entering such default values temporarily for a series of records. In the Data Entry Services menu (EXE1) option D 'Define default values' can

be invoked, and the default values may be entered in the worksheet. For example, the Conference name, place, date which will recur in the bibliographic details of all the papers of a Conference preceedings selected for cataloguing/indexing; or title of a serial from which several papers are catalogued/indexed etc. Using option C 'Clear default values' in the same menu these temporary default values can be deleted from the worksheet when no more required. In the same way again typing errors in data entry can be minimized.

However, there have been very few attempts at providing online facility at the input and processing stage. In 1987 Khalid Basher Mohammed developed two CDS-ISIS Pascal programs to assist at data entry stage in MicroIsis (version 1). QUICK.PAS quickly searches, retrieves and opens data entry worksheets for corrections and modifications. The capability of the program has been subsequently enhanced by Srilatha and Aparna Shridhar (1993). The second program MTHES.PAS, a modification of the THES.PAS program provided by Unesco along with the standard MicroIsis, enables selection of terms from a thesaurus and add it to a specified field in a MicroIsis database record. The program, however, has not been made available widely for testing. F.J. Devadason (University of Papua New Guinea) developed a tutorial package using a story-board like software which is intended for use as

a CAI (Computer Aided Instruction) facility for MicroIsis. The International Centre for Living Aquatic Resources Management (ICLARM) in Manila developed a program to assist in formulating display specifications (ICLARM Personal Communication). Several CDS-ISIS Pascal programs are available, again to assist in the design and development of databases, data entry, global editing of records, generation of reports, etc. these are: NEWDB, GDE, GMOD, GLB, QUICK and their modified versions as mentioned above (Srilatha and Aparna Shidhar 1993). Since the current MicroIsis version 3.0 and above support networking, i.e. permits interaction of several end-users from different terminals, it is essential to provide online facility to ensure quality of input and therefore of the output of information system to its clients.

1.2.3 Problems of Quality of Input Data

A typical data processing procedure consists of three phases - input, processing and output (Davis 1969, 1981). Conventionally data is keyed into the computer by data entry operators or read from one or more input devices; the data is then processed and the results are written to one or more output devices. Input of data is the first phase in a data processing chain (combined functions of input, processing and output), and it is

very important at this stage to make sure that the input is correctly entered into the system regardless of the input method used.

Quality cannot be ensured consistently in the final products and services that clients receive unless appropriate quality is ensured in the inputs and processes that ultimately lead to the products and services (that is, quality should be ensured in every component and process of information resource management). Quality is not accidental, but must be deliberately built into the system.

There are a few reasons why errors occur in the input data. These are:

- data may be incorrectly recorded from its source;
- data may be corrupted, lost etc. when converting from one format to another;
- data may be incorrectly read and therefore incorrectly entered into the computer; and
- data may be lost in handling.

With the view to ensuring quality of inputs, a number of factors need to be taken into consideration during data input preparation. Some of these factors are:

- Skills and knowledge of data entry personnel;
- Capability of the computer system in terms of user-friendliness which could promote and provide easy to learn and easy to understand

techniques to all categories of users (e.g. data entry personnel) ranging from novice (including occasional and new system users) to expert ones; and

- Capability of the computer system in terms of storage space, speed, data maintenance, data security and availability of in-built consistency checks.

In developing countries, Africa in particular, it is most likely that the type of computer operators (data entry operators) will be those with only a basic knowledge of computer operations. Indeed, in libraries and information centre environments a computer-based information system operator may have only a minimum knowledge of information technology. Therefore, the input preparation, including data entry and ensuring consistency and standardized ways of rendering names of personal authors, corporate authors, conferences, title of serials, etc. based on appropriate guidelines, codes, authority lists, etc. should be made easy and error-free.

The attention being paid in research and studies on information system design and development, to improving the quality of services and information system output products is understandable. However, it needs to be reiterated that the quality of system output depends on the quality of the data input, its preparation and

processing. Unless these activities also benefit from the developments in information technology, quality of output products and services and their speed of delivery cannot be expected to improve to any significant extent.

From the preceding discussions it is apparent that, the input phase is a crucial phase but also the weakest phase and link in any data processing chain.

A number of features and capabilities are being incorporated into software packages and the related documentation by software developers in an attempt to securing quality of input and reducing error rates which could impair the quality of products and services rendered. Some of these features include:

- providing input data entry manuals, help messages, printed authority lists, and data validation checks;
- provision of expanded echo of input data to provide opportunity for visual verification; and
- presentation of sufficient information on output for recipient to verify processing and other control checks.

Since data does exist in different types namely numeric, textual, graphical, etc. the in-built house-keeping routines of a software may not guarantee quality of inputs to the system. These features are capable of monitoring range limits in the case of numeric data and

not exact data value. Serious problems may be experienced with the systems which totally rely on the above mentioned error control features particularly when entering strings of characters. For example, when corporate author, serial titles, monographic series, class numbers and corresponding subject headings are mishandled, the system will not be able to detect such an error.

Recognizing such problems a number of manuals have been prepared to supplement whatever is available in the database management system (DBMS) software itself, designed to improve quality of input data. Examples are: MIBIS (Microcomputer-based Bibliographic Information Systems), IDIN (International Development Information Network), DIVINER (Water Resources and Sanitation Network for South and Southeast Asia), etc. These are primarily prepared to guide data entry personnel to represent and render the names of subjects, personal authors, corporate authors, etc. according to widely accepted cataloguing and indexing codes. However, when the data is keyed in, typing and spelling errors occur, and the system may not be able to check these, unless someone proof reads the data entered. Online data entry facility will be of help as the data entry personnel need only to automatically download the selected data element(s) to the specific field(s) in a record.

1.2.4 Need for Online Data Validation Facilities

Most of the word processing software have in-built procedures which normally assist users in a number of ways (Chaudhry 1989). Some of these features include:

- Spelling check and thesaurus control;
- Print enhancements; and
- Page formatting facilities.

In the case of DBMS and text retrieval software we need also vocabulary control tools for use in providing standard ways of entering different data items such as corporate names, serial titles, monographic series, personal names, class numbers, descriptors, subject headings, based on agreed and established codes and rules, e.g. Anglo-American Cataloguing Rules (AACR2), International Standard Bibliographic Description (ISBD), etc. Also, given the complexity of MicroIsis formatting language it would be useful to have an online help on format errors to assist users on how to handle such errors occurring during the preparation of display/output format. These utilities will help to improve the quality of input data before saving into the MicroIsis databases and later being submitted for further processing.

Ganzmann (1990:153) indicated that in an integrated system the function of control of input must be supported by the vocabulary control software as well, that is the

program must check whether a term used for indexing is contained in the electronic thesaurus or not. This criterion can be applied to the DBMS and text retrieval software. Certain features which determine the structure and complexity of the formatting language, of the authority list(s) and other features related to consistency control basing on the established codes and rules is desirable.

To achieve the above mentioned objectives of improving quality of information services, an online data validation user-friendly system which could meet the needs of different categories of users is desirable. The notion of user-friendliness is associated with systems which could accommodate all categories of users in terms of their ease of learning, ease of understanding and ease of use. The Dictionary of Computers and New Information Technology defines user-friendliness "as a system with which relatively untrained users can interact easily" (cited in Perera 1992). Haneock-Beaulieu (1992) pointed out that user-friendliness is a notion which emerged following the advent of microcomputers and spread of information technology in many different types of uses. Kemp indicated that "user-friendliness is not an absolute concept but a relative one, it relates to the individual user" and that "user-friendliness for interfaces as a whole and for their various components, is achieved when

inexperienced users find that they make the systems to which they are attached simple and easy to use and when experienced users are able to use them quickly without having to enter redundant information and when they can give complex instructions to do complicated things" (cited in Perera 1992). To the user of whatever category, the system is represented primarily by the interface. A complex interface reflects a complex system to the user; a simple interface suggests a simple system to the user.

The conclusion to be drawn from this is that user-friendliness is not an inherent characteristic of the software but a way in which each individual user experiences it.

Users fall into three classes depending on their experience of using the software: the novice, the intermediate and the experienced. Generally a system can be called user-friendly if it satisfies all these categories of users by providing unambiguous instructions at different interaction levels for them. MicroIsis scores highly on the part of system prompts and scores poorly in terms of handling error message and help message (Perera 1992). Perera (1992) concluded her evaluation of MicroIsis by saying that "Micro CDS-ISIS does not provide user instructions properly and the limited amount of instructions provided are not sufficient even for the intermediate and experienced

users."

The use of MicroIsis in Africa and other parts of the world has increased significantly (PADIS 1988, 1990; International Classification (1993)). There is thus a need to examine the problem and provide solution. As already indicated, much work has been accomplished in providing user interface in the search and retrieval phase. More recently Srilata and Aparna Shidhar (1993) developed a CDS-ISIS Pascal program (RECOM.PAS) for checking duplicates in data entry in MicroIsis database especially if the input records are received on diskettes from different sources. The program provides also for consistency check for forms of data entry e.g. Personal authors, Corporate and Conference, class numbers, etc.

As already mentioned in sections 1.2.1 and 1.2.2 the problem of lack of online data entry guide and help on format errors does exist with the current MicroIsis software which limits significantly the possibility of achieving the aim of having high quality of input into the databases built using MicroIsis. The researcher of this project feels that online data entry guide and help on format errors will be of much help in improving the quality of information output through improving quality of input data by:

- Improving speed and efficiency of data entry;
- Minimizing the possibility of inconsistency in

data entry in the different fields; and

- Optimizing the effective use of the system.

The present work therefore attempts to develop some utilities for an online data validation facility for MicroIsis aimed at improving the quality of information services rendered using the software.

1.3 SCOPE AND LIMITATION OF THE STUDY

The scope of this study is restricted to the development of online data validation facilities (online data entry guide and help on format errors) on databases built using MicroIsis software package. The utility programs developed take into consideration the interest of different categories of users (novice, intermediate and experienced) of the software.

However, the lack of access to the MicroIsis source code to a large extent hampered the possibility of developing a program on format errors which could capture and locate the error position in order to facilitate and speed up corrections.

Since the operating system commands may be different from one computer system to another, the utilities need some functions before running them successfully. But the utilities can run successfully in VAX versions and on an IBM PC and compatible micros.

The types of users for whom these utilities are designed are data entry operators, library and information professionals using MicroIsis software. The utilities are developed using CDS-ISIS Pascal language to provide online data validation facilities. The program for data entry guide will require the data entry operator to press a function key indicated at appropriate points in the data entry worksheet to activate it and proceed as guided by the program. Similarly a function key is to be pressed while in 'Browse' or 'Display search results' of the Information Retrieval Services to execute the program on format errors.

The present work used MicroIsis versions 3.03, 3.07 and the associated CDS-ISIS Pascal language. However, with some modifications these utilities could successfully be run in MicroIsis version 2.3.

1.4 SIGNIFICANCE OF THE STUDY

Data validation in data processing environment is essential in order to ensure high quality of input into the system aimed at ensuring high quality of information output to the clients.

In Africa national and international institutions, whose activities require inter-alia development and maintenance of bibliographic, referral and textual

databases using MicroIsis software will benefit from this project. In Africa, it is estimated that there are over 500 users of the software.

In general, this study will benefit MicroIsis users elsewhere as well by introducing new features which are not available with the standard MicroIsis software. Further, the study could contribute to the improvement of future versions of the MicroIsis software.

1.5 METHODOLOGY

Through literature survey of documents relating to MicroIsis software it was established that the problem of lack of online data validation facilities does exist with the software. As mentioned earlier, most of the utilities developed are concerned with assisting users on how to design and develop databases and how to formulate search expressions to facilitate retrieval.

To solve this problem, CDS-ISIS Pascal programs have been developed to interface with the MicroIsis software to provide online data entry guide and online help on format errors.

For the purpose of testing and demonstration of data entry assistance, the data entry guide to the ABNCD database (accommodating bibliographic records, as well as records of profiles of experts, institutions, projects

databases using MicroIsis software will benefit from this project. In Africa, it is estimated that there are over 500 users of the software.

In general, this study will benefit MicroIsis users elsewhere as well by introducing new features which are not available with the standard MicroIsis software. Further, the study could contribute to the improvement of future versions of the MicroIsis software.

1.5 METHODOLOGY

Through literature survey of documents relating to MicroIsis software it was established that the problem of lack of online data validation facilities does exist with the software. As mentioned earlier, most of the utilities developed are concerned with assisting users on how to design and develop databases and how to formulate search expressions to facilitate retrieval.

To solve this problem, CDS-ISIS Pascal programs have been developed to interface with the MicroIsis software to provide online data entry guide and online help on format errors.

For the purpose of testing and demonstration of data entry assistance, the data entry guide to the ABNCD database (accommodating bibliographic records, as well as records of profiles of experts, institutions, projects

and information systems), is used.

To achieve the formulated specific objectives a number of CDS-ISIS Pascal programs have been developed. The CDS-ISIS Pascal programming language of MicroIsis versions 3.03 and 3.07 is used in the development of these programs.

Help and similar guidelines have been incorporated in the utilities so as to enhance the friendliness of MicroIsis software package in general.

The program for online data validation facility will be tested by MicroIsis users in some sectors using the software and the feedback will be used in the development of the final program. The utility program will be incorporated in the MicroIsis version 3.07.

Guidelines for implementation and application of the Pascal programs are provided.

CHAPTER 2

MICROISIS SOFTWARE

2.1 HISTORICAL DEVELOPMENT OF MICROISIS

The mainframe version of CDS-ISIS software package was developed by Unesco in the early 1970's by combining features of their Computerized Documentation System (CDS) and the Integrated Set of Information Systems (ISIS) of the International Labour Office (ILO). Because of the extension of the Unesco Integrated Documentation Network to the regional offices and of the increasing need for small libraries and documentation centres in the developing countries to have access to information handling using modern information technology, Unesco through its Division of Library, Archives and Documentation Services (subsequently merged with the Division of the General Information Programme (PGI)) undertook the development of Mini-micro version of CDS-ISIS.

MicroIsis is a computer-based system for information storage, processing and retrieval and is a generalized system designed for the computerized management of structured textual databases with variable or fixed length fields and records.

MicroIsis is distributed by Unesco free of cost to nonprofit making organizations. Because of its good features, together with its development being supported by an international intergovernmental organization, Unesco, the software is now widely used in developing countries and Europe, for the creation and manipulation of structured textual databases. Such databases include bibliographic, referral type, factual and object-oriented databases (Neelameghan 1992). The software is distributed through national or regional agencies designated for the purpose. Where such an agency does not exist, the software can be obtained directly from Unesco. In any case before getting the software, the requesting institution has to enter into an agreement with Unesco for the license to use the software.

MicroIsis is based on the program of the same name developed formerly in 1965 by ILO for use on IBM mainframe computers. Later on ILO passed the mainframe version to the IDRC, Canada. IDRC developed a minicomputer version of ISIS, known as MINISIS, for use on HP 3000 series of minicomputers.

CDS-ISIS was then given to Unesco who developed it exclusively. Unesco developed a microcomputer version of MicroIsis to respond to the needs of information systems which have already started using microcomputers following the advent and wider availability of microcomputers in

the 1980's. Version 1 was released for general use in December 1985; version 2.3 in March 1989; modified version 2.32 in September 1989 and version 3.0 late 1992; versions 3.03 and 3.07 in 1993.

2.2 GENERAL FEATURES OF MICROISIS

This section describes the general features of MicroIsis software package.

The features of MicroIsis currently (up to version 3.07) applying are as follows:

TABLE 2-1: SYSTEM RESTRICTIONS IN EFFECT IN VERSION 3.07

Data Items	Limitations
Maximum number of databases	Unlimited
Maximum number of records in database	16 million
Maximum record size	8000B
Maximum number of fields(defined in FDT) (excluding repetitions of repeatable fields)	200
Maximum number of FST lines	200
Maximum field size	8000B
Maximum number of fields in a worksheet page	19
Maximum number of pages in a worksheet	20
Maximum size of a display format	8000B*
Maximum number of stopword in stopword file	799
Maximum number of identifiers in an ISIS Pascal program	200
Maximum number of instructions in an ISIS Pascal program (including all programs called in by a uses statements)	10000
Maximum number of loaded programs	10
Maximum number or real constants	200
Maximum Pascal Run-time stack	2000
ISIS Pascal dynamic string area	16932
Worksheet work area size	3KB
Maximum Hit record size	4000*
Maximum number of characters per field	8000B
Maximum number of indexes	1
Maximum number of indexed characters per term	30
Maximum number of literals	132
Maximum number of sort keys	256
Maximum number of characters in search expression	250
Maximum number for a field tag	32767
Maximum number of printed characters in a single page	300
Maximum length of pattern type field	20B
Maximum field length in a worksheet	8000B
Maximum length of the name of a database, display format, or worksheet	6B
Maximum length of a field name in FDT	30B

Notes: * Changes in the current MicroIsis software.

TABLE 2-1: SYSTEM RESTRICTIONS IN EFFECT IN VERSION 3.07

Data Items	Limitations
Maximum number of databases	Unlimited
Maximum number of records in database	16 million
Maximum record size	8000B
Maximum number of fields(defined in FDT) (excluding repetitions of repeatable fields)	200
Maximum number of FST lines	200
Maximum field size	8000B
Maximum number of fields in a worksheet page	19
Maximum number of pages in a worksheet	20
Maximum size of a display format	8000B*
Maximum number of stopword in stopword file	799
Maximum number of identifiers in an ISIS Pascal program	200
Maximum number of instructions in an ISIS Pascal program (including all programs called in by a uses statements)	10000
Maximum number of loaded programs	10
Maximum number or real constants	200
Maximum Pascal Run-time stack	2000
ISIS Pascal dynamic string area	16932
Worksheet work area size	3KB
Maximum Hit record size	4000*
Maximum number of characters per field	8000B
Maximum number of indexes	1
Maximum number of indexed characters per term	30
Maximum number of literals	132
Maximum number of sort keys	256
Maximum number of characters in search expression	250
Maximum number for a field tag	32767
Maximum number of printed characters in a single page	300
Maximum length of pattern type field	20B
Maximum field length in a worksheet	8000B
Maximum length of the name of a database, display format, or worksheet	6B
Maximum length of a field name in FDT	30B

Notes: * Changes in the current MicroIsis software.

As already mentioned, MicroIsis is a DBMS sotfware for information processing, storage, and retrieval on microcomputer. It is a generalized system designed for the computerized management of structured textual databases. It can be implemented on an IBM PC and compatible microcomputers with 512K (preferably 640K) RAM, a floppy disk drive and a hard disk. The powerful indexing/search languages and database import/export using ISO-2709 data exchange format are noteworthy features of the software. Programs are available for down-loading structured texts in ASCII format, and dBASE, FOXBASE and INMAGIC databases. Facilities for handling text in Amharic, Arabic, Chinese, and languages/scripts of East Europe and of India are available, in addition to Latin/Roman scripts. MicroIsis enables the user to build, manipulate, maintain, manage, and retrieve records from, such databases. In particular, it enables the:

- defining of databases containing user selected data elements;
- updating (that is, modification, correction, and deletion) of records in a database;
- automatic creation of fast access files e.g. inverted/index file for any or all of the fields (data elements) in a record in each data base using nine indexing techniques;
- retrieval of records from one or more databases

using simple or complex search expressions such as Boolean logic, adjacency, greater than, less than, equal to, not equal to and other operators.

- displaying the list of terms in the index file facilitating the selection of terms to formulate search expressions;
- data may be retrieved by using inverted files or by searching the records sequentially (also called "free text searching");
- displaying the number of hits (postings and records) for each component of the search expression;
- re-execution of an earlier search expression in the same or another database resident in the same directory;
- displaying of records in accordance with user defined display formats;
- sorting of retrieved records in any desired sequence(s);
- printing out entire database or retrieved records partially or all of them and/or indexes;
- exchanging or merging of records of two or more databases that are in compatible formats (e.g. ISO-2709 format);
- filing order may be changed through the use of special characters. This means, for instance, that

an article at the beginning of a title can be ignored when the title field is being sorted to create a title index and a title beginning with a digit may be filed as if the number were written as a word; and

- enhancing the capabilities of the system, such as, interfacing a thesaurus, or the IDAMS software for statistical work, using programs written in CDS-ISIS Pascal.

A number of new features and facilities such as, Local Area Network (LAN) support which were not available in version 2.3 have been introduced in version 3.0. The LAN has been successfully tested with Novell Netware, 3COM and Banyan Vines. This facility enables concurrent access to a database by several users for both searching and data entry. Certain functions, such as, master file backup and restore, inverted file update, modifying records, and import operations, may only be performed by one user at a time. New indexing techniques, conversion of a Hit file to a master file, expanded memory manager (EMM), new print formatting parameters, CDS-ISIS Pascal procedures for arrays of strings and some other new features are supported.

MicroIsis provides additional features: numerical functions: RAVR, RSUM, RMAX, RMIN and VAL; string functions: REF and REF(L) for linking up records resident

in the same database. It also provides facility of repeatability of fields and breaking fields into sub-fields.

The current version of MicroIsis has either added to or modified some of the CDS-ISIS Pascal procedures and functions (see chapter 3). A new print option has been added to allow produce ASCII files with no carriage returns other than the ones inserted through the new line commands of user defined format. To use this option, enter 'A' in the 'first page number' field of the print worksheet. With version 3.07, it is now possible to produce print files in Microsoft Rich Text Format (RTF), which may later be printed using a Word processor accepting RTF (such as Microsoft Word or Word for Windows). An optional translate table, ISISLP.TAB, may now be provided to customize the character codes for printing. This table is built in the same way as ISISUC.TAB and must be stored in the ISIS menu sub-directory (as specified in parameter 2 of SYSPAR.PAR). MicroIsis checks the presence of this table and if found, it will use it to translate the data in the print file.

In order to enhance the capability of the MicroIsis expanded memory support has been provided which is equally well adopted in version 3.07. This facility allows the user to increase the number of instructions in CDS-ISIS Pascal, CDS-ISIS Pascal Run-time stack and CDS-

ISIS Pascal string work area. In version 3.07 the problem experienced by most of the users when using 192K of expanded memory and printing a sort report with a format containing CDS-ISIS Pascal format exit of not properly printing the first record has been fixed.

Generally, MicroIsis allows to build and manage an unlimited number of structured textual databases. The databases can be manipulated in a different number of ways as explained in section 2.2 above.

The operation of MicroIsis software can be performed through the main menu (see Figure 2-1) by typing a single character corresponding to the desired function. On selecting the menu option system may display another menu from which further options may be selected or request the user for additional data to complete the selection. MicroIsis is a multilingual software, hence menus and prompts may be displayed in any of the languages adopted.

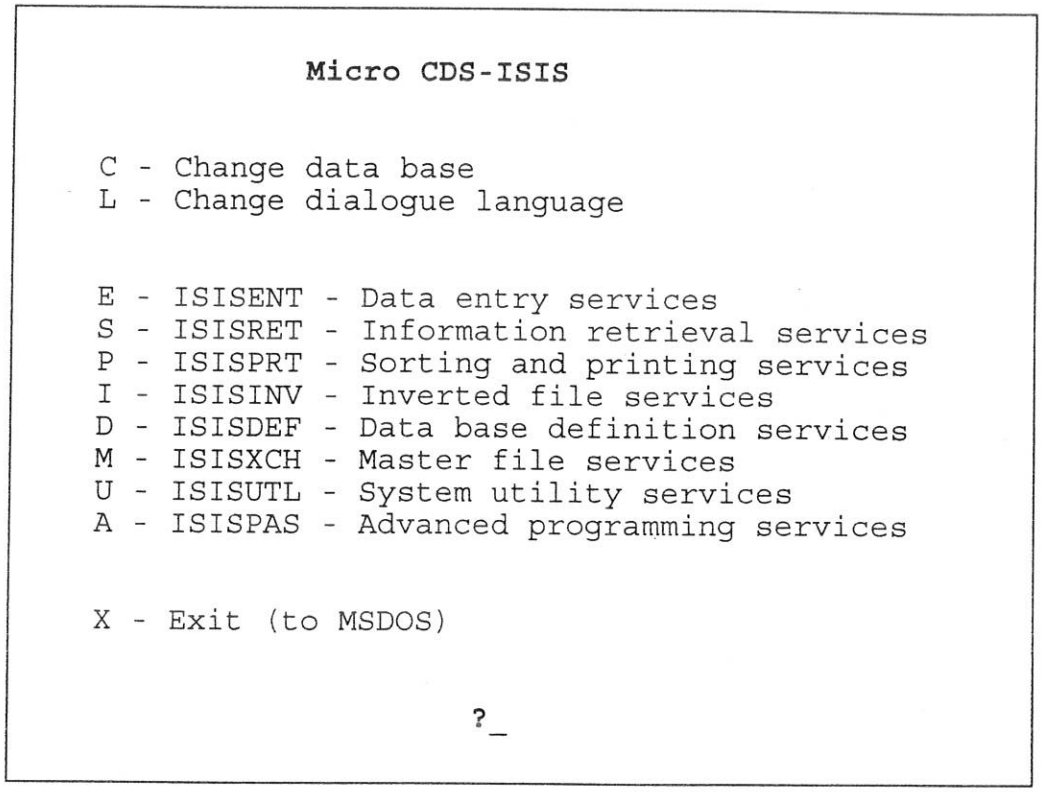


Figure 2-1: CDS-ISIS main menu

2.2.1 Local area network (LAN) support

The versions (3.0 and above) of MicroIsis provides simultaneous access to a given database by two or more users for both searching and data entry in a LAN environment on Novell Netware and other network software. However, certain functions, such as master file backup/restore, inverted file update or import operations may only be performed by single user at a time. MicroIsis provides appropriate locks to prevent this to happen. An attempt to use one of these functions while the database is locked by another user, MicroIsis will cause

appropriate messages and cancel the request. Likewise, the system will not allow the modification of a record which is already being modified by another user at the same time. To operate in LAN mode, parameter 14 in SYSPAR.PAR or parameter 0 of dbn.PAR should be set to appropriate value (see section 2.5.2 (n)).

2.2.2 Expanded memory support and system restrictions

Expanded Memory Manager (EMM) is now supported if installed. EMM is used to expand the amount of memory available to the software. It is possible to limit if necessary the amount of expanded memory to be used by MicroIsis through parameter 13 of SYSPAR.PAR as follows:

13 = 0 no EMM will be used

13 = 64K at most 64K of EMM will be used

13 = 128K at most 128K of EMM will be used

13 = 192K up to 192K of EMM will be used (default).

The system parameters affected by EMM are as follows:

Data item	With EMM	Without EMM
Maximum number of identifiers in an ISIS Pascal program	200	200
Maximum number of instructions in an ISIS Pascal program (including all programs called in by a USES statement)	16383	10000
Maximum number of loaded programs	10	10
Maximum number of real constants	200	200
ISIS Pascal Run-time stack	16000	2000
ISIS Pascal dynamic string area	49500	16932
Format work area	32767	8000

Table 2-2: System restrictions affected by EMM.

2.2.3 Indexing techniques in FST

There are nine indexing techniques currently available from which user can choose.

Indexing technique 0 is used to build an element from each line. Here the whole field is indexed, but as lines. User should specify MHL mode in data extraction format.

Indexing technique 1 is used to build an element from each sub-field and/or line extracted by the format.

Indexing technique 2 builds an element from each term or phrase enclosed in tringular bracket <>.

Indexing technique 3 does the same as 2 but the term or phrase is enclosed between slashes //. These slashes are always shown in the display.

Indexing technique 4 is used to build an element from each word in the text. In this technique if the field contains sub-field delimiters then MHL must be specified.

In addition, indexing techniques 5, 6, 7 and 8 are used to specify a prefix for search terms extracted using indexing techniques 1, 2, 3 and 4. The prefix is specified in the data extraction format as an unconditional literal as follows:

'dp...pd', [format]

where d is a delimiter of user choice (which does not occur in the prefix); 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=.

2.3. MICROISIS FILES

MicroIsis consists of a number of files which can be categorized into two based on their purpose. These are: system files and data base files. The system files are those common to all MicroIsis users whereas data base files are not. The naming conventions and the file

extensions used in different files of the system is described below. All MicroIsis files have standard names as follows:

nnnnnn.eee

where:

nnnnnn is the file name (all file names of MicroIsis, except for the names of programs are limited to a maximum of six characters)

.eee is the file extension identifying a particular type of file.

2.3.1 System files

MicroIsis system files are those files which include the various executable programs as well as system menus, worksheets and message files provided by Unesco together with those developed by MicroIsis users.

i. CDS-ISIS Program

The name of the MicroIsis executable program file as supplied by Unesco is **ISIS.EXE**. Depending on the release and/or target machine, there may also be one or more overlay files. These files have file extension **OVL**.

ii. System menus and worksheets

All system menus and worksheets of MicroIsis have extension **FMT** and the names are created as follows:

pctnnn.FMT

where:

p is the page number assigned automatically by the system (such as **A** for the first page, **B** for the second page, **C** for the third page, etc).

c is the language code (e.g. **E** for English), which must be one of those provided for in the language section menu **xXLNG**

t is **X** for menus and **Y** for system worksheet

nnn is a unique identifier.

For example the full name of the English version of the menu xXGEN is AEXGEN.FMT.

The system automatically inserts the file extension as well as the page number. Therefore, when MicroIsis prompts for a menu or worksheet name, the user need only to provide the name with neither its extension nor its page number as the system assigns these automatically. Usually menus and worksheets are restricted to one page.

The language code is mandatory for the system menus and worksheets. For example if user wants to link a HELP menu to the system menu EXGEN, its name must begin with the letter E. Since X and Y are used to enforce system

menus and worksheets respectively, it is a good practice not to include any of these two letters in the second position of the data base name, failure of which the corresponding data entry worksheets will be created in the system worksheet data base directory (parameter 2 of SYSPAR.PAR). Although this will not prevent normal operation of the data base, it is not recommended.

iii. System message files

All system messages and prompts are stored in standard MicroIsis databases. All corresponding database files are needed when updating a message file, but only the master file is used to display messages. MicroIsis has a message database corresponding to each language supported through the language selection menu xXLGN. The database name assigned to message database is xMSG (where x stands for language code).

iv. System tables

MicroIsis makes use of its three system tables currently available to define character sets. These are: ISISLP.TAB (used for customizing character codes for printing), ISISUC.TAB (used for defining lower to upper-case translation) and ISISAC.TAB (used for defining the

alphabetical characters). ISISLP.TAB was not available in previous versions (up to 3.0).

v. System print and work files

Some MicroIsis print functions do not send the output directly to the printer but store it on a disk file from which one may then print. These files have file extension **LST** and are reused each time the corresponding function is executed.

In addition, MicroIsis creates temporary work files which are automatically discarded at the end of the MicroIsis session. These temporary files can be retrieved in the next session in case the previous session was not terminated normally (e.g. session termination due to power failure). Work files have file extension **TMP**.

Some of the print and temporary work files produced by MicroIsis are:

IFLIST.LST .... Inverted file listing file (produced by ISISINV)

WSLIST.LST .... Worsheet/menu listing file (produced by ISISUTL)

xMSG.LST ..... Sytem messages listing file (produced by ISISUTL)

x.LST ..... Printed output (produced by ISISPRT when printing no print file name is supplied)

SORT10.TMP .... Sort work file 1

SORT11.TMP .... Sort work file 2

SORT12.TMP .... Sort work file 3

SORT13.TMP .... Sort work file 4

SORT20.TMP .... Sort work file 5

SORT21.TMP .... Sort work file 6

SORT22.TMP .... Sort work file 7

SORT23.TMP .... Sort work file 8

TRACE.TMP ..... Trace file created by certain programs

ATSF.TMP ..... Temporary storage for hit lists created during retrieval

ATSQ.TMP ..... Temporary storage for search expressions.

2.3.2 Database files

A MicroIsis database consists of a number of physically distinct files. These files can be classified into three (mandatory, auxiliary and user files).

Mandatory database files are those established when database is created by means of ISISDEF services and they are always present.

Auxiliary files are normally created by the system whenever certain functions are performed. For the economy of space these files should be periodically deleted whenever are no longer needed.

User files are those created by the MicroIsis user

(such as display formats), which are fully user determined.

2.3.2.1 Master File

In MicroIsis all records of a given database are stored in the master file. Individual records are identified by a unique number, which is automatically assigned by MicroIsis when they are created. This number is called master file number (MFN). In order to provide fast access to each master file record, MicroIsis associates a special file to the master file called the crossreference file. It is essentially an index to the master file giving the location of each record in the master file. The master file records may be updated (created, modified or deleted) by means of the data entry facilities of the ISISENT services.

The crossreference file, also known as Master file index, is a MicroIsis database file with file extension **XRF**. XRF file is organized as a table of pointers to the Master file. The first pointer corresponds to the MFN 1, the second to MFN 2, etc.

2.3.2.2 Inverted file

In a MicroIsis database a master file record can be

retrieved directly by its master file record number (MFN), through the crossreference file. But to get access to a record in a database by using string of characters (e.g. by author, by title, by subject, etc) rather than by record key, additional ways are of course necessary. MicroIsis provides facility for handling such cases through an inverted file. The inverted file is nothing but an index to the contents of the master file. For instance, three master file records (with MFN 10, 200 and 1030) contain the keyword **ARTIFICIAL INTELLIGENCE**. The logical structure of the corresponding inverted file entry would be:

ARTIFICIAL INTELLIGENCE 10 200 1030

In the above example, **ARTIFICIAL INTELLIGENCE** is the access point (or dictionary term) and each reference to the master file record where it appears is called a *posting*.

In practice each term will have a different number of records indexed under it, hence, the logical records in an inverted file vary in length. Again to provide fast retrieval of each access point, the inverted file consists of six physical files of which five contain dictionary of searchable terms organized in a B*tree. Despite the fact that B*tree is more complex in processing than sequential file processing, it has been adopted in the design of MicroIsis inverted file

structure because of the advantage that B*tree structure does not suffer from the overflow problems. For the purpose of disk optimization, two B*trees have been designed, one for maintaining terms with characters up to 10 (stored in files .N01/.L01) and the other one for terms with characters longer than 10 and up to 30 characters. Further, the file .CNT (control) is developed to store fields to both B*trees mentioned. In each B*tree, the file .N0x contains the nodes of the tree and .L0x file contains the leafs. The leaf records point to the posting file .IFP.

In addition, MicroIsis provides nine indexing techniques (see section 2.2.3) which allow selective creation of inverted file for each database to suit user needs.

The searchable elements for a given database are defined by the user in a field select table (FST), which contains the fields to be inverted and the indexing technique to be used for each field or sub-field and specification of the format for extraction of the data.

2.4 ISO-2709 INTERCHANGE FORMAT

ISO (International Organization for Standardization) format files may be used to produce and convert into MicroIsis format by the ISISXCH program. The ISO record

consists of three logical segments: LEADER (fixed length for every record), DIRECTORY (consists of variable number of fixed-length (12 characters) entries, each giving identification length and location of each variable field in the record, it ends with a separator (#). VARIABLE DATA (the data themselves in the form of variable field length alphanumeric information. Each field ends with a field separator (#).

The leader and directory are both control segments which are used to process data contained in the third segment. A record may be of any length (maximum 8000B). It normally ends with a record terminator (##).

The directory is essentially a record index. For each tag there will be an index, this is why one can have variable number of directories. Each directory consists of a fixed length field, where:

- 3 characters give the tag;
- 4 characters give the length of the variable field; and
- 5 characters give the location of the field, relative to the beginning of the variable fields area.

2.4.1 Format of ISO files produced and accepted by MicroIsis

ISO files produced by MicroIsis are standard text files in the format described in section 2.4. Because it is impractical to handle text files with relatively long lines, especially when they have to be inspected with line editors and/or transmitted over telecommunication lines, MicroIsis will split each record into 80-character line segments, each followed by standard line termination ^M^J (Carriage Return/Line Feed). All segments, except the last, will contain exactly 80-characters of text (see annex 2-1).

2.5 SYSTEM INSTALLATION

Previously it was necessary to create a directory in which ISIS program will be installed. In addition sub-directories (SYS,PROG, MENU, MSG, DATA) were allowed to be created for installing the software. It was also necessary to define the path where the ISIS program is stored to enable MS-DOS to find it. This was done by defining the system files (AUTOEXEC.BAT). A CONFIG.SYS should be created with the lines Buffer=24, files=24. These files can be edited using any text editor, e.g. EDLIN.

In the versions 3.0 and above of MicroIsis a new installation program (ISISINST) is now provided. To install version 3.0x one has to mount diskette ISIS01 and type:

X:isisinst then press <ENTER>

where X is the drive of the diskette.

The program will then provide online guidance through the installation procedure. Further, the old INSTALL.BAT is still provided for manual installation.

2.5.1 Hardware requirements

For the minimum hardware requirements needed to run MicroIsis see section 2.2 above.

2.5.2 SYSPAR.PAR: Global parameters

The system file SYSPAR.PAR contains setup parameters. It is read each time MicroIsis is activated and may be used to override default values which the system would otherwise use. It can be created/edited using any text editor e.g. EDLIN. Each parameter should start in a new line. The general format is as follows:

n=value

where

n is the parameter number (see below); and

value is the corresponding value.

It should be noted that the value must immediately follow the equal sign. Any space after the equal sign and before the value will be taken as part of the value. The parameters which user may specify are described below.

a. Parameter 0: SYSPAR.PAR re-direction

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

- a full file name, which may optionally include drive and/or directory; and
- a question mark (?) or an exclamation mark (!) followed by a prompt, for example:

0=?Please enter name of system parameter

This will cause the system to first display this prompt and then read name of the parameter file from the keyboard. If exclamation mark is used instead of the question mark, the text will not be echoed to the screen. In practice, this is not recommended.

The parameter 0 may be used to provide 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 that the following file has been created.

IBM PC

SYSPAR.PAR	MOHAMED
1=\isis\prog\	4=\isis\mohamed\work\
2=\isis\menu\	5=\isis\mohamed\data\
3=\isis\msg\	6=OMES
7=S	7=E
0=!password	

In this case each time MicroIsis is started it will first read SYSPAR.PAR and set the system, menu and message path and select Spanish as the default language. Then the user will be prompted to enter the password (parameter 0). If the response to this is Mohamed, MicroIsis reads this file and sets the remaining parameters. This user will therefore work in english and use OMES as his default database. To provide certain amount of security user may in addition use appropriate attributes(under MS-DOS) to hide these files.

where the MicroIsis will create any required work file (e.g. sort work file). By default, work files are created in the current directory. All files not covered by other parameters will be assigned to this path.

f. Parameter 5: Data base path

All files with file extension ANY, CNT, IFP, L01, L02, N01, N02, MST, XRF, FDT, FST, FMT, PFT,STW, SRT and other database files such as HIT, LN?, LK?, etc. are assigned to this path.

g. 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.

h. Parameter 7: Default language

This parameter is a one-letter character code specifying the initial dialogue language to be selected. The initial default language is E (English).

l. Parameters 11 and 12: Define graphic characters

These parameters allow to define the graphic characters for boxes of type 1 and type 2 respectively. These parameters only apply to the PC version of the system and consist of a string of six characters defining:

- the vertical sides;
- the horizontal sides;
- the top left corner;
- the top right corner;
- the bottom left corner; and
- the bottom right corner.

m. Parameter 13: Expanded memory support

This parameter has been added to control expanded memory support (PC version only) see section 2.2.2.

n. Parameter 14: Network control

This parameter has been added to control network support. It is defined as follows:

14=0 single user (non-network version);

14=1 full network version - this parameter allows the simultaneous searching and updating of both the

master and inverted file by more than one user.

14=2 restricted network support - this parameter allows the simultaneous searching and updating of the master file by more than one user.

o. **Function key definition parameters**

MicroIsis provides facility for defining function keys to perform automatic typing of often recuring series of keystrokes. For instance, a function key may be defined to perform an entire job such as a print run or an export operation.

2.5.3 dbn.PAR: Database parameters

MicroIsis will normally try to locate a file dbn.PAR (where dbn is the database name) in the database path specified in parameter 5 of SYSPAR.PAR. If this file does not exist, then all the database files are assumed to be in the database path. dbn.PAR allows to define individual paths for specific data base files.

2.6 CONCLUDING REMARKS

The various features of MicroIsis mentioned in section 2.2 and the flexibility of the software in terms

of export/import, availability of powerful indexing techniques and search language, local area network support and given that the development of the system is supported by an international intergovernmental organization, Unesco, and that its usefulness in information storage and retrieval system (ISRS) has been proven in a large number of institutions, small and large, makes it suitable for handling textual data bases in libraries and information centres.

ANNEX 2-1: TRANSLATION AND BREAKDOWN IN ISO-2709

The following example is a record taken from a MIBIS database. The corresponding ISO record generated by MicroIsis is provided (Figure 2) and its breakdown of the codes in ISO-2709 (Figure 3).

Record No. 1

The following is a translation of the record to ISO-2709

02085000000000445000450000500110000000600110001100700020002200800020002400900030002601200030002902000030003202100030003599900020003810000730004011000240011311000150013711100910015211400860024311400980032912100230042712200050045012300110045513000090046613100070047516000140048216100110049620200560050730000450056330100630060830300100067131009020068132000200158340000500160341000004016084100005016124110012016174120002016294200005016314300003016364#1988-07-10#1989-02-23#M#C#CA#24#En#En#B#Manual for the preparation of records in development-information systems#^aMorin-Labatut,Gisèle#^aSly,Maureen#^aInternational Development Research Centre^bInformation Sciences Division^cOttawa, ON^dCA#^aTechnical Meeting on Common Methodologies^bOttawa, ON^cCA^d3-7 Nov1981^e1981-11-03 #^aTechnical Meeting on Common Methodologies^bMont Sainte-Marie, PQ^cCA^d9-13 Nov 1981^e1981-11-09#^aOttawa
, ON^bIDRC^cCA#1982#1982-00-00#^a272 p.#^av.1#0-88936 -354-4#IDRC-TS40e#Recommended methods for development-informationsystems#<MANUALS><CATALOGUING><INFORMATION ANALYSIS>#<DEVSIS><INDEXING><CONTENT ANALYSIS><AGRIS><METHODOLOGY><ISIS>#<MINISIS>#This publication contains guidelines for the record structure and content of bibliographic records in development-information systems. It is intended to
be a guide for the designers of new systems and for those seeking to improve existing systems. The manual is the result of 5-6 years experience with a variety of regional and national DEVSIS (Development Sciences Information System) systems and can be considered to be an update of the technical recommendations of the DEVSIS Study Team in 1975. The major part of the manual contains field-by-field guidelines to be used by a documentalist when creating records in a development-information system. Numerous annexes are appended, including a data-definition table, sample completed worksheets, a correspondence for UNISIST and AGRIS field tags, guidelines for implementation under MINISIS, CDS/ISIS and DOS/ISIS, a glossary, and a bibliography.#Information systems#LIST#REF#MAIN#025.315 MOR#2#MONO#CD##

The following is a breakdown of the codes in ISO-2709

02085000000000445000450 [Leader] 005 [Record tag] 0011 [Length of variable field] 00000 [Beginning of variable field area] 006 [Record tag] 0011 [Length of variable field] 00011 [Beginning of variable field area] 007 [Record tag] 0002 [Length of variable field] 00022 [Beginning of variable field area] 008 [Record tag] 0002 [Length of variable field] 00024 [Beginning of variable field area] 009 [Record tag] 0003 [Length of variable field] 00026 [Beginning of variable field area]
012 [Record tag] 0003 [Length of variable field] 00029 [Beginning of variable field area]
020 [Record tag] 0003 [Length of variable field] 00032 [Beginning of variable field area]
021 [Record tag] 0003 [Length of variable field] 00035 [Beginning of variable field area]
999 [Record tag] 0002 [Length of variable field] 00038 [Beginning of variable field area]
100 [Record tag] 0073 [Length of variable field] 00040 [Beginning of variable field area]
110 [Record tag] 0024 [Length of variable field] 00113 [Beginning of variable field area]
110 [Record tag] 0015 [Length of variable field] 00137 [Beginning of variable field area]
111 [Record tag] 0091 [Length of variable field] 00152 [Beginning of variable field area]
114 [Record tag] 0086 [Length of variable field] 00243 [Beginning of variable field area]
114 [Record tag] 0098 [Length of variable field] 00329 [Beginning of variable field area]
121 [Record tag] 0023 [Length of variable field] 00427 [Beginning of variable field area]
122 [Record tag] 0005 [Length of variable field] 00450 [Beginning of variable field area]
123 [Record tag] 0011 [Length of variable field] 00455 [Beginning of variable field area]
130 [Record tag] 0009 [Length of variable field] 00466 [Beginning of variable field area]
131 [Record tag] 0007 [Length of variable field] 00475 [Beginning of variable field area]
160 [Record tag] 0014 [Length of variable field] 00482 [Beginning of variable field area]
161 [Record tag] 0011 [Length of variable field] 00496 [Beginning of variable field area]
202 [Record tag] 0056 [Length of variable field] 00507 [Beginning of variable field area]
300 [Record tag] 0045 [Length of variable field] 00563 [Beginning of variable field area]
301 [Record tag] 0063 [Length of variable field] 00608 [Beginning of variable field area]
303 [Record tag] 0010 [Length of variable field] 00673 [Beginning of variable field area]
310 [Record tag] 0902 [Length of variable field] 00681 [Beginning of variable field area]
320 [Record tag] 0020 [Length of variable field] 01583 [Beginning of variable field area]
400 [Record tag] 0005 [Length of variable field] 01603 [Beginning of variable field area]
410 [Record tag] 0004 [Length of variable field] 01608 [Beginning of variable field area]
410 [Record tag] 0005 [Length of variable field] 01612 [Beginning of variable field area]
411 [Record tag] 0012 [Length of variable field] 01617 [Beginning of variable field area]
412 [Record tag] 0002 [Length of variable field] 01629 [Beginning of variable field area]
420 [Record tag] 0005 [Length of variable field] 01631 [Beginning of variable field area]
430 [Record tag] 0003 [Length of variable field] 01636 [Beginning of variable field area] #[Field separating character] 1988-07-10#[Field separating character] 1989-02-23#[Field separating character] M#[Field separating character] C#[Field separating character] CA#[Field separating character] 24#[Field separating character] En#[Field separating character] En#[Field separating character] B#[Field separating character] Manual for the preparation of records in development-information systems#[Field separating character]^aMorin-Labatut,Gisèle#[Field separating character]^aSly,Maureen#[Field separating character] ^aInternational Development Research Centre^bInformation Sciences Division^cOttawa, ON^dCA#[Field separating character]^aTechnical Meeting on Common Methodologies^bOttawa, ON^cCA^d3-7 Nov1981^e1981-11-03 #[Field separating character]^aTechnical Meeting on Common Methodologies^bMont Sainte-Marie, PQ^cCA^d9-13 Nov 1981^e1981-11-09#[Field separating character]^aOttawa
, ON^bIDRC^cCA#1982#1982-00-00#[Field separating character]^a272 p.#[Field separating character] ^av.1#[Field separating character]0-88936 -354-4#[Field separating character]IDRC-TS40e#[Field separating character] Recommended methods for

development-informationsystems#[Field separating character]<MANUALS><CATALOGUING><INFORMATION ANALYSIS>#[Field separating character] <DEVSIS><INDEXING><CONTENT ANALYSIS><AGRIS><METHODOLOGY><ISIS>#[Field separating character]<MINISIS>#[Field separating character]This publication contains guidelines for the record structure and content of bibliographic records in development-information systems. It is intended to be a guide for the designers of new systems and for those seeking to improve existing systems. The manual is the result of 5-6 years experience with a variety of regional and national DEVSIS (Development Sciences Information System) systems and can be considered to be an update of the technical recommendations of the DEVSIS Study Team in 1975. The major part of the manual contains field-by-field guidelines to be used by a documentalist when creating records in a development-information system. Numerous annexes are appended, including a data-definition table, sample completed worksheets, a correspondence for UNISIST and AGRIS field tags, guidelines for implementation under MINISIS, CDS/ISIS and DOS/ISIS, a glossary, and a bibliography.#[Field separating character]Information systems#[Field separating character]LIST#[Field separating character]REF#[Field separating character]MAIN#[Field separating character]025.315 MOR#[Field separating character]2#[Field separating character]MONO#[Field separating character]CD##[Record terminator]

CHAPTER 3

CDS-ISIS PASCAL

3.1 INTRODUCTION

This work is concerned with the development of utilities written in CDS-ISIS Pascal language to interface with databases designed and developed using MicroIsis. Therefore, it will be useful to look at the features of CDS-ISIS Pascal language. The programming language used for MicroIsis is largely Assembler, with a small component in Pascal. The standard MicroIsis's capability can be enhanced by programs written in CDS-ISIS Pascal, a subset of standard Pascal with its own library of functions and procedures. CDS-ISIS Pascal programming facility is provided along with the standard MicroIsis.

CDS-ISIS Pascal is based on the standard Pascal programming language. The specificity of CDS-ISIS Pascal is its library of predefined procedures and functions to provide access in a convenient manner to most MicroIsis functions. As indicated in chapter 2 (section 2.3.1), CDS-ISIS Pascal is an integral part of CDS-ISIS and consists of a compiler, an interpreter and a library. The compiler produces a pseudo-code (.PCD file) which is

machine-independent, that is application programs written in CDS-ISIS (for an IBM PC or compatible machine will run without change on a VAX machine). It should be noted that some procedures may operate only with certain type of machines, for example RECORD LOCKING may be used with PC version only. Hence, if portability is desired such procedures should not be used in the program development.

The program written in CDS-ISIS Pascal must be stored in a file with extension PAS, in the program directory as indicated in parameter 1 of SYSPAR.PAR file. To compile a program, option A (Advanced programming services) is to be selected from the MicroIsis main menu. The following prompt will be displayed:

C[ompile] R[un] Q[uit] ?

Then an option C is to be typed followed by the name of the program at the prompt, then press <ENTER> key to complete.

The executable code created by the MicroIsis compiler is stored in a file with extension PCD using the same file name of the source program. Although CDS-ISIS Pascal language is developed using standard PASCAL, it deviates from standard PASCAL in some ways.

Unlike the standard Pascal, the CDS-ISIS Pascal does not support CONST and TYPE declarations and exponent

notation. In addition, DATA functions, end of line (EOLN) function and multiple CASE label constants are not supported. However, end of file (EOF) function, ARRAY of the type REAL or STRING are supported and all variables must be declared in a VAR clause before they are used. All REAL procedure and function parameters are passed by value, while STRING parameters are passed by reference. The VAR parameter prefix, however, should not be declared. Although the compiler does not issue a message if an assignment is made to a formal parameter passed by value, the value of the corresponding actual parameter in the calling (main) procedure will not be affected.

Likewise, the compiler does not check whether the actual parameter, corresponding to a formal parameter passed by reference, is an expression and the procedure assigns a value to it. It is the programmer's responsibility to ensure that whenever a procedure assigns a value to a formal parameter passed by reference, the corresponding actual parameter is always a single variable.

CDS-ISIS Pascal expressions (such as arithmetic, boolean and string) follow standard Pascal syntax, although boolean expressions are supported only in conditional statements. The boolean type variables cannot be declared.

CDS-ISIS Pascal statements are implemented in the

same way as they are in the standard Pascal unless otherwise stated. The CASE selector can either be REAL or STRING expressions. In the case where string expressions are used the corresponding case labels may be strings or single character. However, in order to be matched properly, the expression must yield a string which has the same length as the case label constants. As it was indicated above, multiple case label constants are not supported.

CDS-ISIS Pascal has structured control statements (such as, WHILE, REPEAT, FOR, CASE, and IF-THEN-ELSE) that allow the programmer to write clear and conscise source code with the flow of control processing from top to bottom. In addition, it supports modular programming, in which a large program can be broken up into independent functions and procedures, each one performing a specific task. Modularization greatly assists in breaking down the complex problem into small simple modules (or segments) to facilitate coding. Furthermore, it supports recursive programming in which a procedure may be called by itself within the program.

In addition to the standard INPUT and OUTPUT files, two additional predefined text files are available, these are: INP (for input) and OUT (for output). As in standard PASCAL, INPUT and OUTPUT are automatically assigned to the standard input and output devices. In this case the

system will expect data from the keyboard during execution of the program and output results on the monitor (screen). In MicroIsis a RESET or REWRITE statement is automatically issued from all these files. INP and OUT are assigned by default to the standard input and output devices respectively, but this must be reassigned through ASSIGN procedure.

Unlike the standard PASCAL, CDS-ISIS Pascal supports only comments delimited by braces { }. The alternative delimiters (* *) are not supported.

CDS-ISIS Pascal provides possibility for designing and developing a user exit for specific functions which are not available in the standard MicroIsis software package. A user exit must be declared as an attribute in the program statement. Furthermore, unlike the batch programs, user exits may declare program-level parameters which will be passed to the program by MicroIsis whenever the exit is activated. These parameters are fixed for each type of user exit and are checked by the compiler. There are two types of user exits, these are MENU exit and FORMAT exit.

A menu exit is activated each time a specific option is selected from the system menu. A menu option can be associated with a menu exit by assigning to that option action code E (for more information see section "The Line editor" in the Mini-micro CDS-ISIS Reference Manual).

Format exit is activated each time while processing a MicroIsis display format. It may be used to provide data to the standard formatting routine (see the section "The Formatting Language" in the Mini-micro CDS-ISIS Reference Manual). A FORMAT exit is invoked by coding the name of the program in the user display/output format as shown below:

&name(format)

where:

& identifies this is a format exit invocation;

name is the name of CDS-ISIS Pascal program to be executed; and

format is a MicroIsis format

Furthermore, a CDS-ISIS Pascal program may be used to call other CDS-ISIS Pascal programs through the USES statement. This facility enables users to run several programs by invoking one program.

3.2 CDS-ISIS PASCAL LANGUAGE COMPOSITION

As already mentioned, MicroIsis is developed largely using ASSEMBLER (approximately 80 per cent) and the remaining part of approximately 20 per cent standard PASCAL language. Its performance from the user's viewpoint is quite good especially in terms of

operational speed. Since the speed of execution of any software depends very much on the type of algorithm used, programming language(s) used, backing storage and input/output operation, MicroIsis proved to be amongst those text retrieval software packages with almost a zero wait state in retrieval processes.

One of the limitations with the current CDS-ISIS Pascal is that it does not have its own window for entering a source program. Like other utilities developed basing on standard PASCAL such as TURBO PASCAL, it just provides a line command where programs developed using CDS-ISIS Pascal can be compiled, run and quit the advanced programming facility.

The CDS-ISIS Pascal library contains a collection of predefined procedures and functions which can be classified into two broad categories: general procedures and CDS-ISIS procedures. The following pre-defined procedures and functions are currently (up to version 3.07) available. The procedures and functions with an asterisk are those modified or added in the MicroIsis version 3.0 and above.

1. General Procedure and Functions

a. Screen management

ATTR	Set screen attribute
BOX	Draw box
CHATTR	Change screen attribute
CLEARBOX	Clear box
CLEARDATA	Clear data area (lines 1-21)
CLEARLN	Clear line
CLEARMSG	Clear message area (lines 22-24)
CLEAR	Clear screen
COPYSCR*	Copy content of a screen
CURSOR	Set cursor position
GETCC	Get cursor position (column)
GETCL	Get cursor position (line)
PAGE	Select active page
SAVESCR	Save active page
SCREEN*	Store content of COPYSCR in buffer
SCROLL*	Scroll window (box)

b. Keyboard management

AUTOTYPE	Simulated keyboard input
INKEY	Single character input (with echo)
KBDKEY	Single character input (without echo)
DEFKEY*	Define function key
KEYDEF*	Get key definition value
KEYPRESS*	Get keyboard status

c. Type conversion

CHR	Convert real to character
ENCINT	Convert real to string (integer)
ENCREAL	Convert real to string
ORD	Convert character to real
VAL	Convert string to real

d. String utilities

POSITION	Find position of string
SIZE	Size of string
SUBSTR	Substring
UC	Convert to upper case
COPYSTR*	Copy string to another

e. Miscellaneous

ASSIGN	Assign file name
DATESTAMP	Get date and time
FILEXIST	Check existence of file
EXEC	Execute another program
SYSTEM*	Execute VAX/VMS or DOS command

2. CDS-ISIS procedures and functions

a. System

LANG	Current language
MENU	Display system menu
MSGTEXT	Get system message
MSG	Display system message
PATH	Get path defined in SYSPAR.PAR or dbn.PAR
SETLANG	Set dialogue language
SYSVARS*	Modify or display system settings
SUBMENU*	Get submenu internal option identifier

b. Data base

CLOSE	Close data base
DBN	Current data base name
LOCK	Lock data base (VAX only)
MAXMFN	Next MFN to be assigned
OPEN	Open data base
UNLOCK	Unlock data base (VAX only)

c. Master file

CREATE	Create new record (interactive)
DELETE	Delete record
FIELDN	Find field number
FIELD	Get field contents
FLDADD	Add field
FLDCHA	Change field
FLDEL	Delete field
FLDMOD	Modify field
FLDREP	Replace field
MODIFY	Modify record structure
FIELDS	Number of fields in record
FLDTAG*	Get field tag
GETMFN*	Get master file number (MFN)
RLOCK*	Lock master file record
RUNLOCK*	Unlock master file record
NOCC	Number of occurrences of field
RECORD	Get master file record
UPDATE	Update record
OCCSEP*	Get repeatable field separator
SAVEREC*	Same as UPDATE, except that it preserves the record lock if present

d. Inverted file

DELTERM	Delete dictionary term
FIND	Get dictionary term
NXTPOST	Get next posting
NXTTERM	Get next dictionary term
POSTING	Get posting
UPDIF	Update inverted file
NPOSTS*	Get number of postings for a dictionary term
SHOWDICT*	Display search term dictionary
RECUPDIF*	Update inverted file only for record MFN
UNDO*	Restores record to MFN to the status corresponding to the inverted file

e. Searching

NXTPOS	Get MFN of next record retrieved
SEARCH*	Execute a CDS-ISIS search expression
SETPOS	Get MFN of retrieved record
MAXSET*	Get number of search expressions submitted
RECALL*	Get previous search expression component
FLUSH*	Delete search expressions submitted

f. Formatting

FMTNAME	Current format name
FORMAT	Execute display format
GETFMT	Define display format
LINES	Number of lines produced by FORMAT
NXTLINE	Get next line produced by FORMAT
COPYFMT*	Copy current format

g. Editing and data entry

DATAENTRY	Edit a worksheet page
DUMMYREC	Establish a dummy record
EDIT	Edit a string through the CDS-ISIS field editor
WORKSHEET	Select worksheet
EDITMOD*	Get edit mode
WSNAME*	Get current worksheet name

For more information consult Mini-micro CDS-ISIS Reference manual.

CHAPTER 4

DATA ENTRY ASSISTANCE

(DEA)

4.1 GENERAL INTRODUCTION

As discussed in section 1.2.3 quality of information products and services that clients receive cannot be ensured consistently unless some aspects of the quality have been ensured in the inputs and processes that ultimately lead to these products and services. It was also mentioned in section 1.2.2 that with the view of minimizing typing errors in the data entry into databases, the standard MicroIsis software package provides facility in the data entry services for defining default data values in one or more fields in the record of a database.

This chapter "Data Entry Assistance" (DEA) presents a CDS-ISIS Pascal program facility for online data entry assistance. The DEA.PAS program is written in CDS-ISIS Pascal langauge, modifying and integrating the following programs:

- GDE.PAS program which was originally developed by Andràs Szücs, Database Manager, UN ACCIS, Geneva and can be used for editing, inserting fields after/before, cancel changes and exit, delete

fields, change field tags, save and exit. The modified version enables entering new records and provides online field-by-field guide to databases created using ABNCD format; and

- VOCON.PAS program developed by Gobinda G. Chowdhury, A. Neelameghan and Sudatta Chowdhury, at SISA. This program can be used, among other things for online selection of term(s) and/or code(s) from a vocabulary control tool, such as, a thesaurus, subject headings list, classification scheme, nomenclature list or code list that is in the form of a MicroIsis database, and adding the selected term(s) and/or codes to specified field(s) in the record(s) of a MicroIsis database, either replacing the existing data in those fields or adding to the existing data.

This chapter is divided into two parts: part one discusses online data entry guide to databases created using MicroIsis and part two deals with online use of authority files for certain data elements (e.g. corporate bodies, personal authors, title of serials, etc.). The separate parts are discussed in sections 4.3 and 4.4 respectively.

4.2 IMPLEMENTATION OF DEA.PAS PROGRAM

The program DEA.PAS program should be copied to the

PROG subdirectory of the ISIS directory or as indicated in parameter 1 of SYSPAR.PAR file. The program should be compiled to generate the compiled version (DEA.PCD) file by selecting option A in the MicroIsis main menu (see Figure 2-1), then option C (compile) in the sub-menu, typing in the program name DEA and pressing the <ENTER> key.

The program can then be interfaced with and called from different points as described below:

- Using the Advanced Programming Services in the main menu, then selecting option R[un] in the submenu, typing in the program name DEA and pressing the <ENTER> key. The following screen will be displayed:

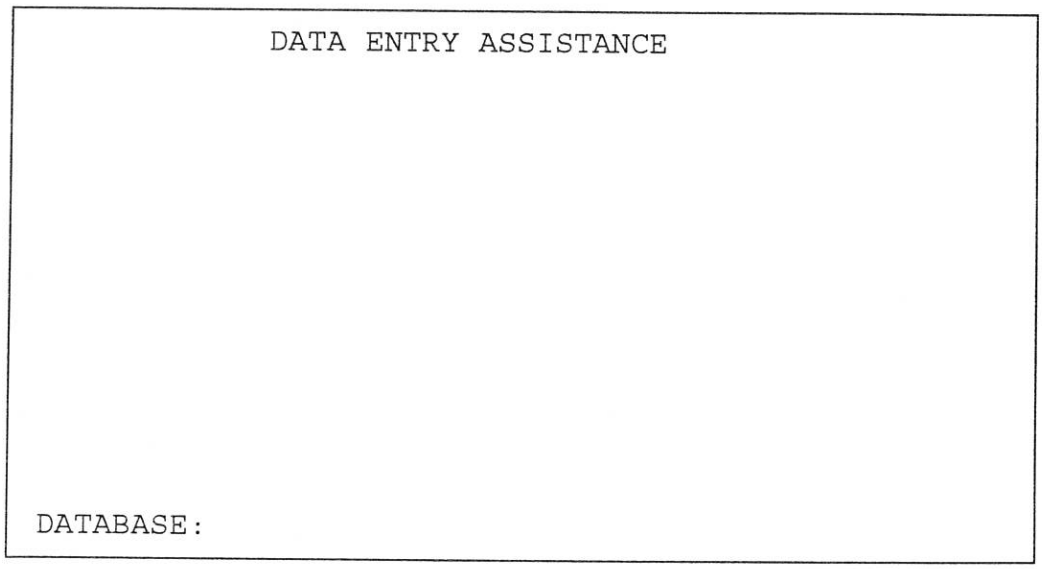


Figure 4-1: Data entry assistance screen 1

Pressing <ENTER> key after keying in the valid MicroIsis database name, the following screen will be

displayed:

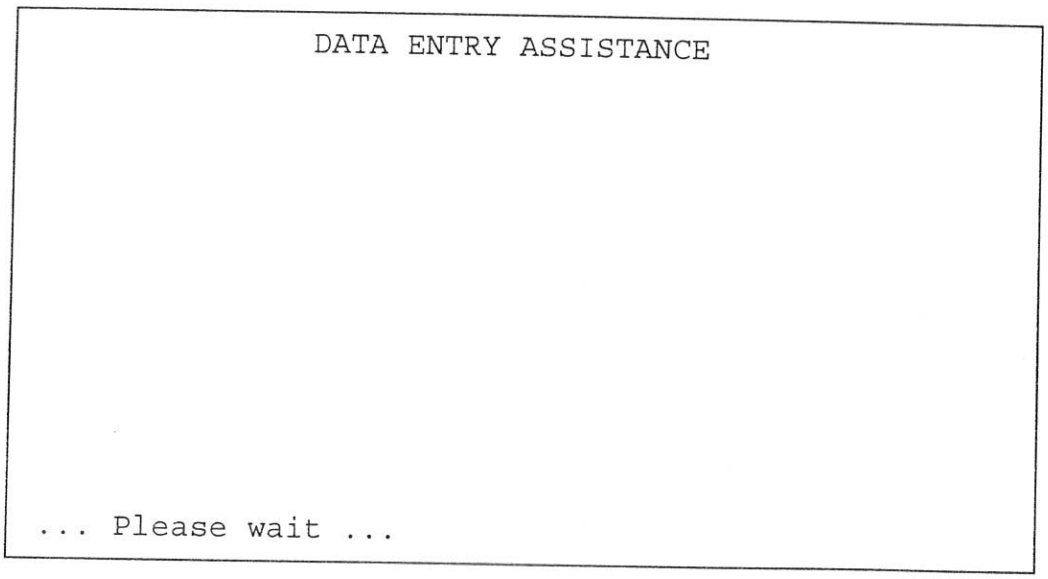


Figure 4-2: Data entry assistance screen 2.

- Modifying the Data Entry menu EXE1 (using option U - System Utility Services in the MicroIsis main menu), introducing a new line, say, A - Data Entry Assistance, and interfacing DEA program at option A in this submenu (EXE1). Then pressing A in menu EXE1 will execute this program.
- While entering data in a record (using either option E for editing an existing record or option N for creating a new record in a menu EXE1 Data Entry services), pressing <PgDn> and <Alt-F1> keys as indicated at the bottom of the worksheet screen, will execute DEA program. The <PgDn> and <Alt-F1> option has been introduced in the submenu of data entry worksheet.

- Interfacing DEA program at option E for editing an existing record and option N for creating new record in the Data Entry menu (EXE1) may cause the program be called whenever either option E or N is pressed.

The DEA program can be used with any of the GUIDE databases or authority databases created using MicroIsis. The databases should be copied to the directory inc by parameter 5 of SYSPAR.PAR file.

4.3 PART I: ONLINE DATA ENTRY GUIDE TO DATABASES

4.3.1 Introduction

MicroIsis is a menu-driven software package allowing its users to create and manage structured textual databases, that is databases whose major constituents is text. MicroIsis through its Data Entry Services (ISISENT) provides all functions relating to data entry operations. However, it does not provide an adequate online data entry guide facility to make it convenient to its users. Perera (1992) points out this problem by saying that "Micro CDS-ISIS is friendly to experts but hostile to novice."

The guide could be a facility to provide online guidance to data entry operators using appropriate database guides in machine readable form. This type of

help facility could be of assistance to all categories of MicroIsis users. Perera (1992) asserts that:

> "The naive user especially cannot make good use of the system unless the context-sensitive help messages are provided, but this does not mean that they are required only by the naive users, even the intermediate and experienced users sometimes require them."

Ganzmann (1990:153) mentions clearly that integrated thesaurus modules in retrieval programs demand specific functions in order to gain an advantage over the printed thesaurus. Equally well DBMS and text retrieval software need specific functions to provide users with online access to manuals currently accessible in printed form. Although some of these manuals such as ABNCD is now available in machine readable form, there is no program developed to enable the ABNCD guide to be displayed online at appropriate points providing guidance to data entry operators. The primary aim of these manuals is to provide guidance to data entry operators in rendering various data elements in the respective fields in a record. Hence, providing these guides in machine-readable form will facilitate ease of access than consulting them manually.

Therefore, this work is designed for the purpose by developing a CDS-ISIS Pascal program (DEA) to assist data entry operators by providing online field-by-field guide

at the input stage aimed at minimizing errors usually occurring during data input.

4.3.2 Using Online Data Entry Guide

Normally database designers do include brief help in the worksheet to provide assistance to data entry operators at the input stage, on the selected information extracted from the guide. For instance, in MicroIsis, for displaying such help messages, the function key <F1> may be pressed when cursor is moved to a field for entering data. The help facility provided through the function key <F1> may have two limitations:

- Some institutions may not have experienced or even intermediate MicroIsis user and information professionals capable of applying various norms and rules established for record structures as stipulated in the manuals correctly, to properly guide data entry operators which may result in inconsistency in rendering data elements in the respective field(s) in a record; and
- In the case where a data entry operator needs more information about a particular field or fields in a record, this help facility may not be adequate.

Therefore, data entry operators need guidance to correctly handle different data elements in a field in a record of a database. For instance, a guide may provide

the following information for each field: field tag, field name, applicable to, available worksheet for that field, repeatability, subfields (SB), representation, example on how to enter data, whether indexed or not, index technique used if any, whole or SB and remarks.

Remarks may be used to provide more information on whether the field is to be used when the document is part of serials or monographic series, etc.

It is known that some information which may be included in the monographs may not be needed in the journal articles. Hence, when using default worksheet of a large database having say 100 fields or more, e.g. ABNCD or MIBIS database, several <ENTER> keys may be pressed before reaching the desired field. This may in practice slow down the data entry operation. In addition to providing online field-by-field guide to data entry operators, DEA program enables MicroIsis to directly call a particular field in a particular master file record in which edit/modify operation need to be done. This speeds up the operation of data entry by avoiding pressing several <ENTER> keys as may be the case when using default worksheet of a large database.

While in the Data Entry menu (EXE1), pressing option E for editing an existing record will execute the DEA program. The Data Entry Assistance Screen 2 shown in figure 4-2 will be presented. The message 'Please wait' will be displayed allerting the user to wait while the

program is openning the database specified earlier. Then the Data entry assistance screen 3 (see Fig. 4-3) will be presented. The name of the current database will be shown at the top left corner of the screen. In example ABNCD and the heading of the DEA program at the middle 'Data Entry Assistance'.

Pressing option X at this point, the program (DEA) will return back to the Data Entry Services menu (EXE1) of the MicroIsis.

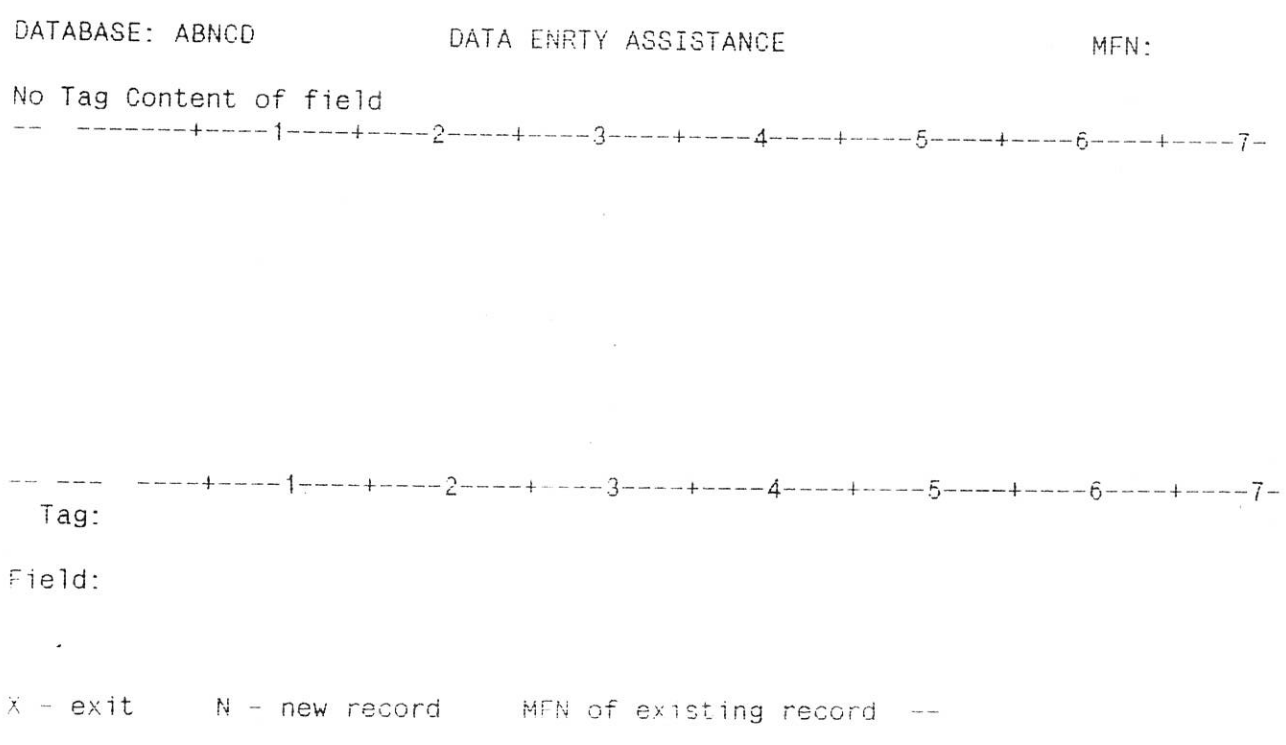


Figure 4-3: Data entry assistance screen 3.

Keying in the existing master file number (MFN) of the current database, say 82, and press <ENTER> key the program will display the contents of the master file number 82 as shown in figure 4-4.

Figure 4-4: Data entry assistance screen 4.

```
DATABASE: ABNCD          DATA ENRTY ASSISTANCE          MFN: 000082

No Tag Content of field
-- --- ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
01 111 University of Dar-es-Salaam^bTraditional Medicine Research Unit^cDar-es
02 110 ^aBlake, William^bed.%^aArnold, Mathew^bed.

-- --- ----+----1----+----2----+----3----+----4----+----5----+----6----+----7
 Tag:
Field:

F1 Lists FDT Fields, F2 Lists Functions.  NEXT FUNCTION:
```

Function keys <F1> and <F2> may be used to display available fields in the database and functions offered by the program (DEA) respectively.

Pressing <F1> the program will open a window to display the available fields. The window is capable of accommodating twelve fields at one time. To display more fields use the function keys indicated in the cue. User may return to the 'Data Entry Assistance Screen 4' by pressing any other key as shown in figure 4-5.

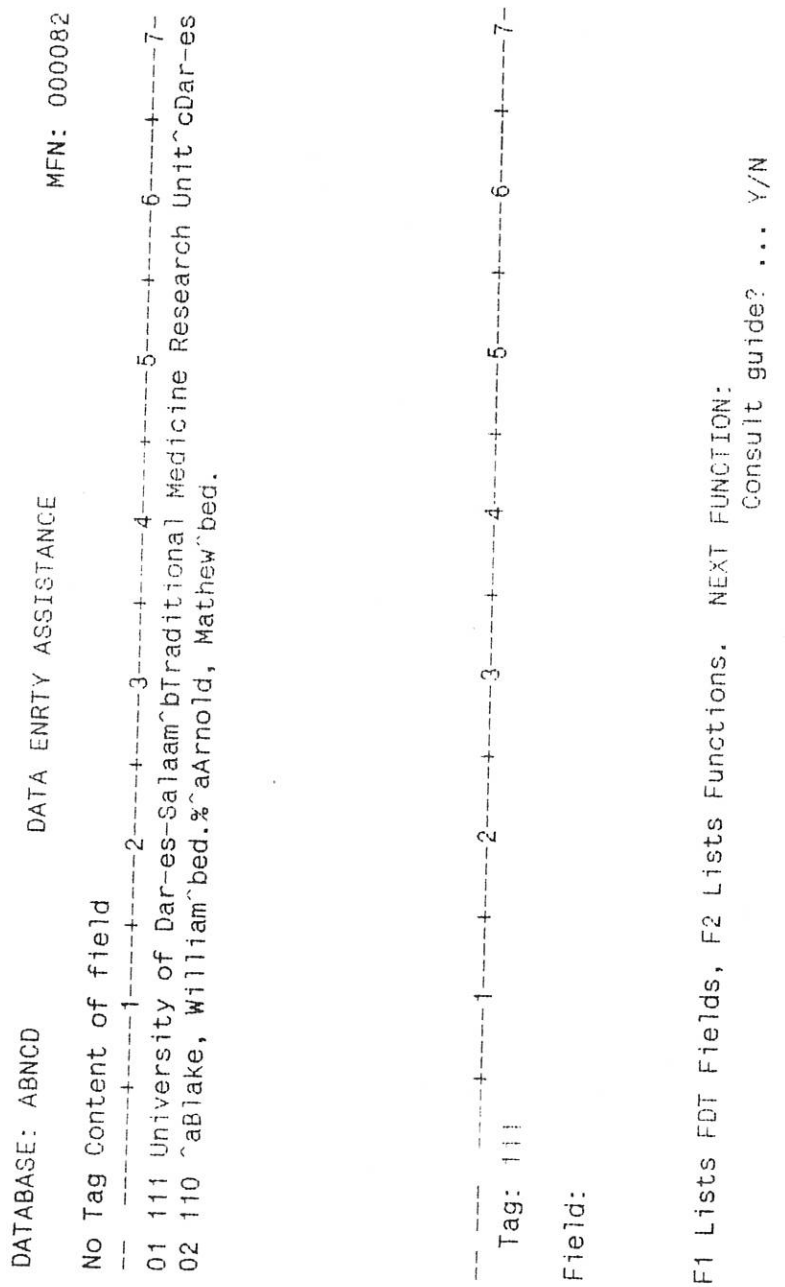
```
Database:  ABNCD
No Tag Content of Field          DATA ENTRY ASSISTANCE                    MFN: 000082
-- ----+----1----+----2----+----3----  Tag          Name of Field
                                       ---    -------------------------------
                                        1     Participating centre code
                                        2     Participating centre record no
                                        3     Record status
                                        5     Date record entered
                                        6     Date record changed
                                        7     Bibliographic level
                                        8     Bibliographic level - parent
                                        9     Country of origin of record
                                       10     Record number of parent
                                       11     Record number(s) of part(s)
-- ----+----1----+----2----+----3----  12     Record no of other lang ver(s)
  Tag: -----                           20     Language of analysis
Field: -------------------------------
-----------------------------------------------------------------------------
-----------------------------------------------------------------------------
Scroll with PgDn, PgUp, ↓,↑ and Home;  other keys to return
```

Figure 4-5: Data entry assistance screen 5.

Pressing function key <F2> another screen window showing the available functions will be presented as in figure 4-6. Pressing any key other than the ones indicated will return back to figure 4-4.

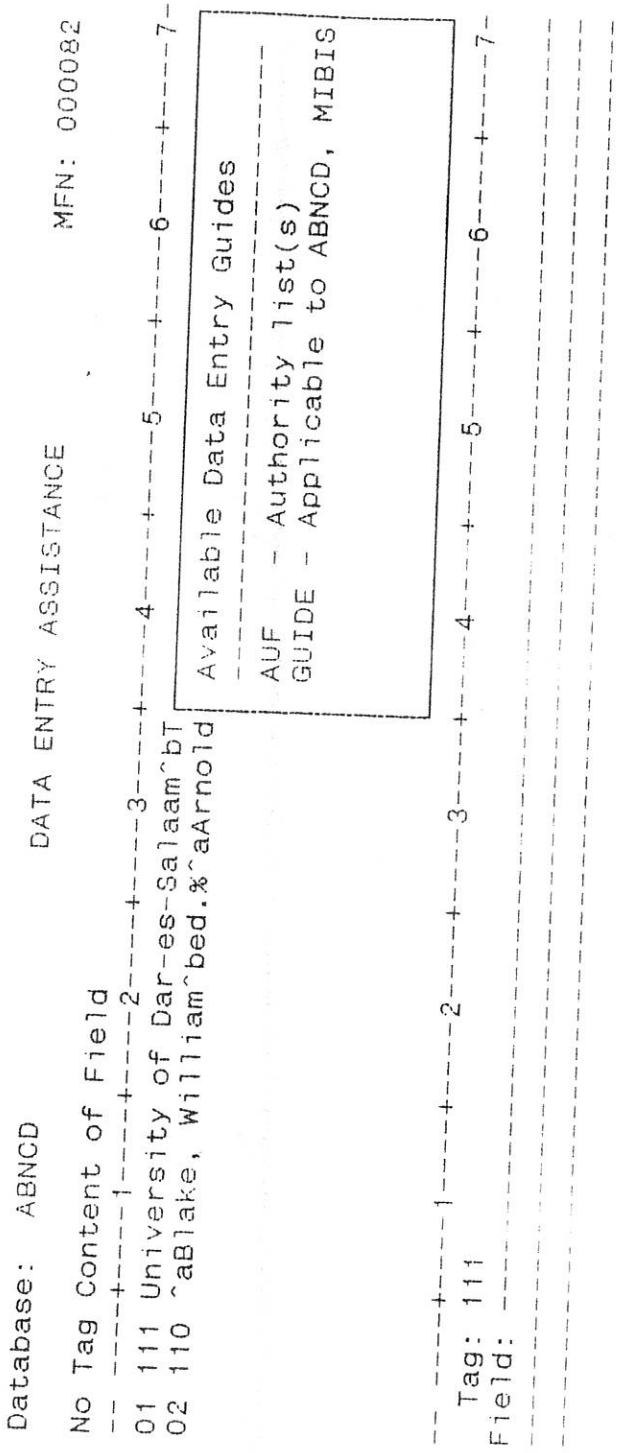
Pressing option E while cursor is in the field 111 will cause the field tag (111) to be displayed at the term 'Tag: ' and the screen shown in Figure 4-7 will be presented.

Figure 4-7: Data entry assistance screen 7.



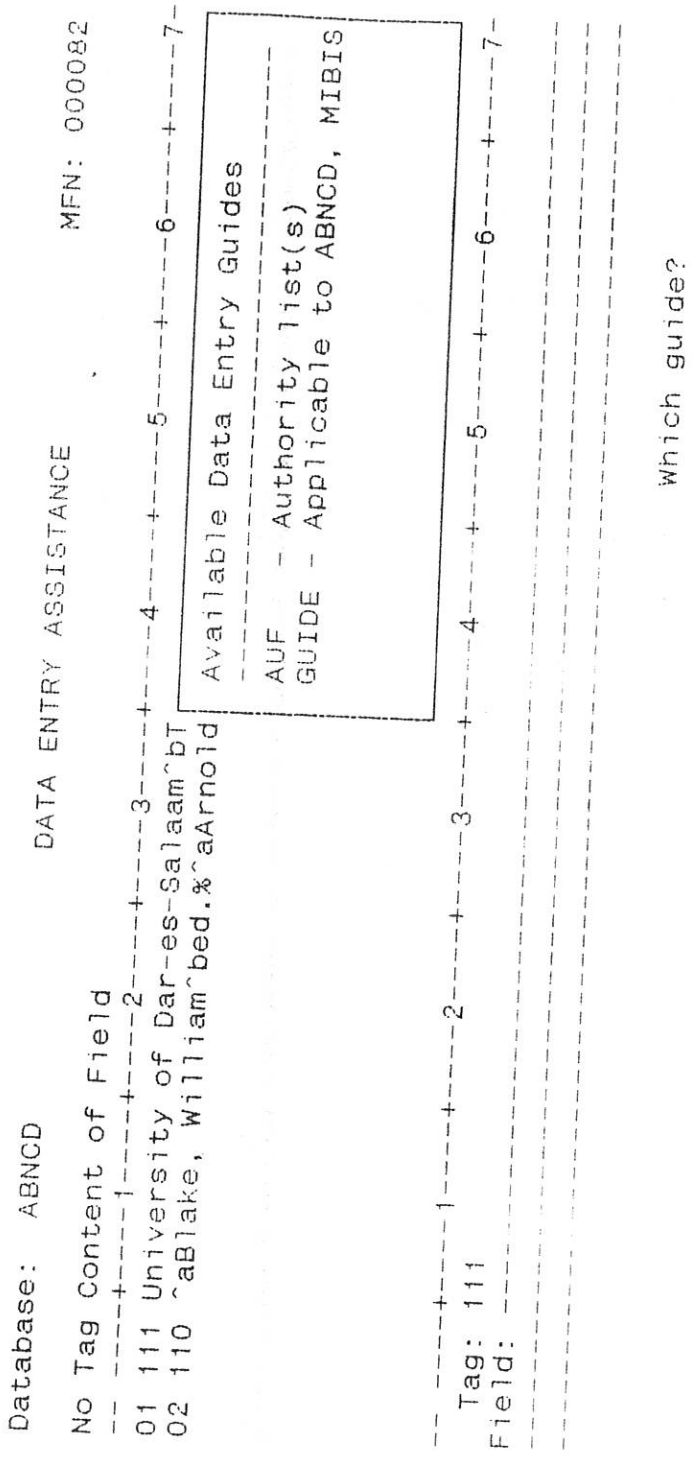
If the response to the question 'Consult guide?' is Y, figure 4-8 will be displayed.

Figure 4-8: Data Entry Assistance Screen 8.



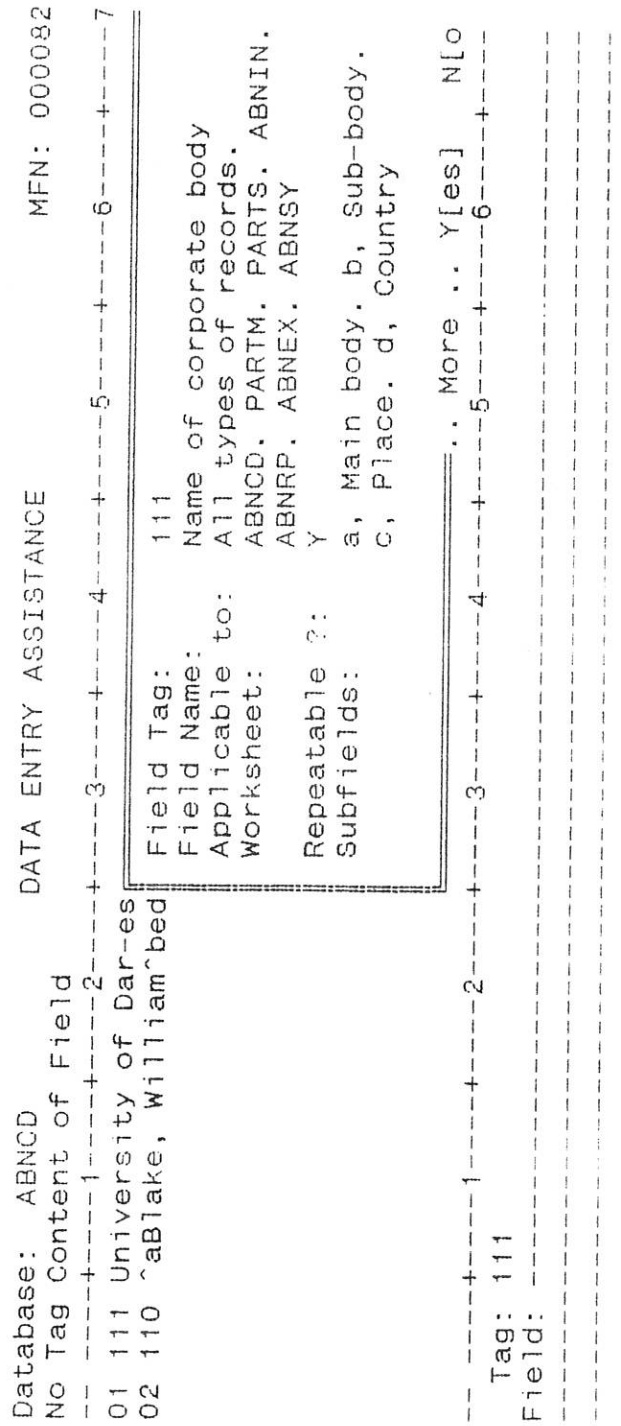
If the response to the question 'Consult guide?' is Y, figure 4-8 will be displayed.

Figure 4-8: Data Entry Assistance Screen 8.



Keying in the name of the guide, say, GUIDE, then pressing <ENTER> key, the program (DEA) will present online field-by-field guide on the field specified earlier (see Figure 4-9).

Figure 4-9: Data Entry Assistance Screen 9.



One may get more information by scrolling through the screen using letter Y as indicated in the lower right corner of the screen window. Once end of record is reached and letter Y pressed or pressing letter N either before calling the guide (see Figure 4-7) or within guide, the program will retrieve the contents of the field at the term 'Field: ' ready for editing (see Figure 4-10). In case INSERT mode is desired as in this example, the INSERT is to be pressed to toggle from REPLACE mode. The missing '^a' can now be inserted before the term 'University'.

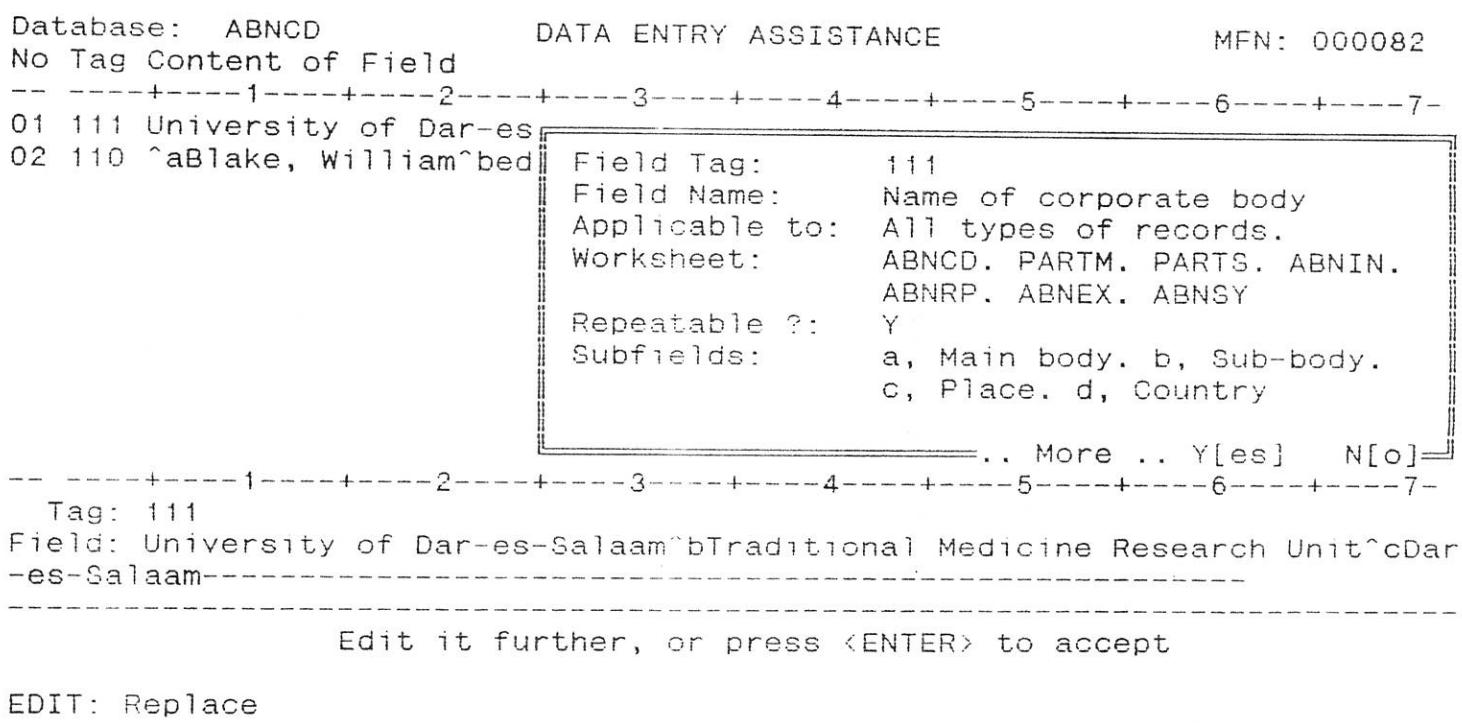


Figure 4-10: Data Entry Assistance Screen 10.

Once editing is over, press <ENTER> key, the contents of the modified field will be displayed as shown in figure 4-11 for visual verification.

The process may be repeated as many times as necessary and once it is over, option X may be used to save changes and exit. The program (DEA) will then return back to the Data Entry Services menu (EXE1).

Option letter C may be used to cancel any changes made in the current session and retain the old version, option D may be used to delete existing field contents and F is used for changing the current field tag.

Similarly, data may be inserted either after or before the current occurrence of the field using options A or B respectively.

A procedure similar to that described above may be used with a new record that is, in the menu EXE1 and in the screen displayed in figure 4-4.

```
DATABASE: ABNCD              DATA ENRTY ASSISTANCE              MFN: 000082

No Tag Content of field
--  -------+----1----+----2----+----3----+----4----+----5----+----6----+----7-
01 111 ^aUniversity of Dar-es-Salaam^bTraditional Medicine Research Unit^cDar-
02 110 ^aBlake, William^bed.%^aArnold, Mathew^bed.




-- ---  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
  Tag: _____

Field: ____________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________

F1 Lists FDT Fields, F2 Lists Functions.  NEXT FUNCTION:
```

Figure 4-11: Data Entry Assistance Screen 11.

In the system where DEA program has been interfaced using only A option, function keys <PgDn> and <Alt-F1> may be used to invoke the program (DEA) and similar procedures described above may be followed.

As mentioned earlier, for the purpose of testing and demonstration of data entry assistance, the data entry guide to the ABNCD database is used. ABNCD is a prototype database for an integrated information storage and retrieval system. The ABNCD database model has been adapted and used in a few operational information storage and retrieval systems. The database(s) developed for these systems have common fields and data elements; there are also some variations among them. The data entry guide used in this thesis is for the ABNCD database model adapted for the DIVINER network (a water and sanitation information network in Asia). The ABNCD database can accommodate the following types of records:

- Documents (i.e. bibliographical records) of various types: books, reports, conference proceedings, analytic of monographs, analytic of serials;
- Profiles of corporate bodies (Institution);
- Profiles of research projects (Project);
- Profiles of persons (Person); and
- Profiles of information systems and services (System).

The data elements (fields) necessary to record

information i.e. to describe the entities mentioned above are defined in a single Field Definition Table of the ABNCD.FDT.

The following example shows the kind of information given for each field.

```
FIELD TAG.......: 140
FIELD NAME......: Monographic series
APPLICABLE TO...: Document. Project.
WORKSHEET.......: ABNCD. PARTM. PARTS. ABNRP
REPEATABLE?.....: Y
SUBFIELDS(SB)...: A, Series title. B, Series part.
REPRESENTATION..: According to cataloguing rules (MIBIS)
EXAMPLE.........: Bibliographical record:
                  ^aOccasional paper, British Museum^bno.
                  33
                  Research project record:
                  ^aSteroid series
INDEXED?........: Y
INDEX TECHNIQUE.: 4
WHOLE OR SB.....: ^a
REMARKS.........: This field should be used when the
                  document or the project is part of a
                  series. MIBIS suggests maintaining an
                    authority file for title of series.
```

For more information about ABNCD database guide and ABNCD.FDT see annexes 4-1 and 4-2 to this chapter.

4.4 PART II: AUTHORITY FILES

Authority files or lists provide a means of maintaining the accuracy and consistency of certain data elements. These files serve to establish the approved forms of names of corporate bodies or other access points, such as names of personal authors, series, subject headings etc. on how they should be entered in a bibliographic record(s).

Normally for each access point an authority record will provide among other things the cross-references and any necessary notes. New records being added to the file must be checked against the authority file to ensure that only one form of name or term is used. Hence, authority file acts in very much the same way as a thesaurus.

Manually maintaining authority files for big libraries or large information systems can be time-consuming and labour intensive task as it entails constant checking for currency and accuracy of records.

Some of the advantages of using computerized authority files are saving time as a record need to be entered only once, reducing the possibility of error, and ensuring consistency (for more information refer MIBIS Manual) (Lauro 1990). As the exchange of bibliographic data in machine-readable form has become more commonplace, the need for computerized authority records has become increasingly important for harmonizing and promoting shared cataloguing.

Suppose the display of record number 155 of the CDS database is as shown below:

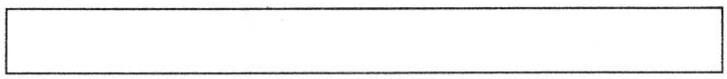
0155 - African development in the 1990's

***** End of display *****

Figure 4-12: Display of Record number 155

Suppose that the name of the corporate body, say SAREC, need to be entered in to the field 71 of record

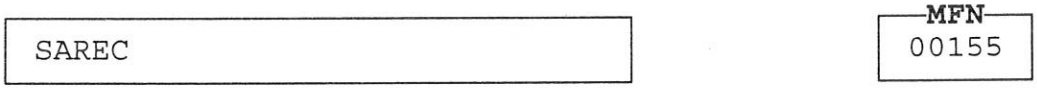
155. Assuming that the data entry operator is not conversant with the form in which the name of the corporate body SAREC is to be entered, he/she may consult the appropriate Guide by following the steps described in section 4.3.2. While in 'Data Entry Assistance Screen 8' (see Figure 4-8). To select AUF (Authority List(s)) key in AUF and press <ENTER> key. Then the screen shown in figure 4-13 will be presented.



Select term

Figure 4-13: Initial Screen for Authority list(s)

If the term selected and entered in the box is 'SAREC' the following screen shown in figure 4-14 will be presented.



- SAREC
- UF Swedish Agency for Research Cooperation with Developing Countries

<SpB>[Next] B[ack] F[irst] P[age] S[elect] T[erm select] Q[uery] M[ark term(S)] ?[Disp query] A[dd rel] D[elete] C[reate term] W[rite in Fld] X[exit]

Figure 4-14: Display of Authority Record.

Other options perform the same functions as defined for the THES.PAS program provided by Unesco.

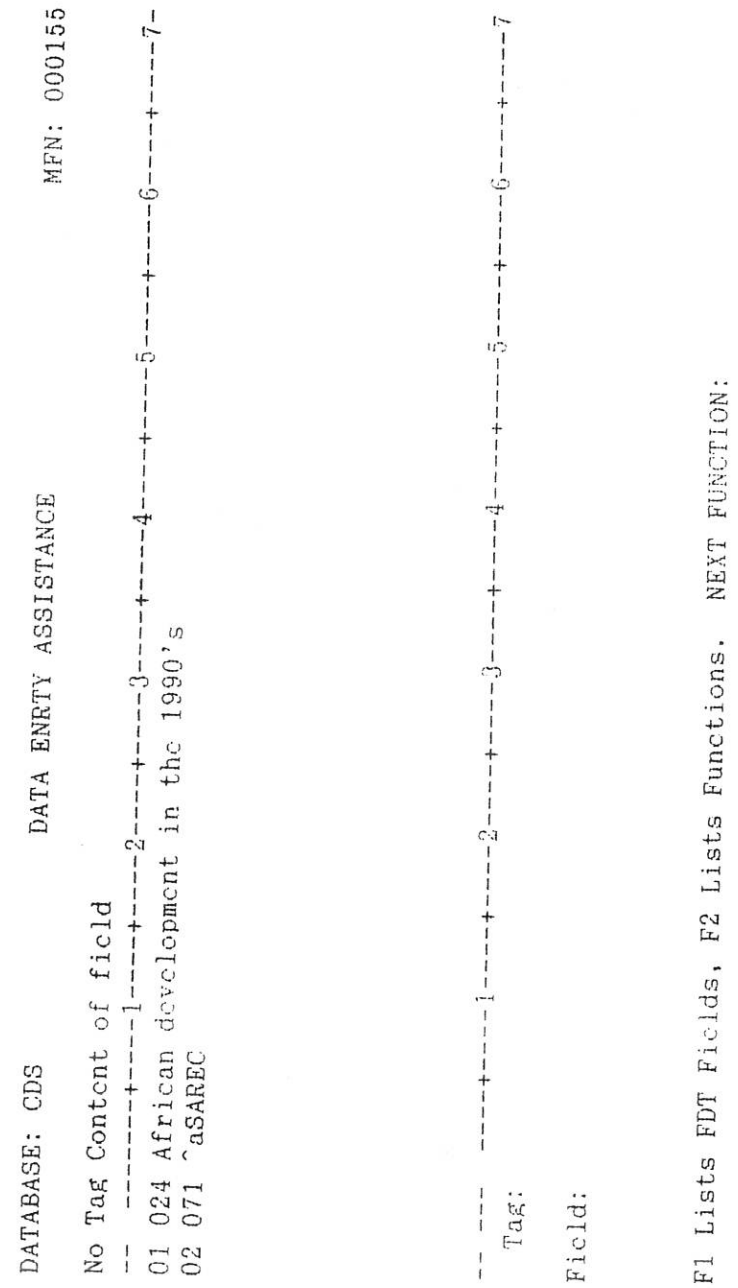
Thus to select term(s) to be added to a field in the the record of a database, the space bar may be used to move the cursor against the desired term(s), (SAREC in this case), and M (mark term(s)) pressed. To write the term(s) on to the record, W (write Fld) is to be pressed. The program will display the following message:

```
Term(s) selected:  ^aSAREC

Do you want to continue -- Y or N: Y
```

If the response is Y, the following message will be presented: In the example (p. 97) keying in 155 and 71 for MFN and field respectively, then pressing <ENTER> key as shown above, the program (DEA) will temporarily store the term 'SAREC' in record number 155, field 71 and return to 'Data entry assistance screen' with record 155 displayed as shown in figure 4-15.

```
Which Master file number (MFN):  155

Which Field:  71
```

Figure 4-15: Record 155 with field 71 entered.



The process may be repeated as many times as necessary. Pressing X will return to 'Data Entry Services menu (EXE1)' of MicroIsis. If record 155 is now displayed, it will be as shown in figure 4-16.

0155 - African development in the 1990's

SAREC

***** End of display *****

Figure 4-16: Record 155 Modified

If the response is N to the query 'Do you want to continue -- Y or N: '(p. 96), the program will again present 'Initial Screen for Authority list(s)' (see Figure 4-13). Data Entry Assistance (DEA.PAS) source code is given in annex 4-3 to this chapter.

Annex 4-1

GUIDE TO DATA ENTRY IN ABNCD DATABASE

Field Tag:	001
Field Name:	Participating centre acronym
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY.
Repeatable ? :	N
Subfields:	None
Representation:	An acronym for each centre participating in the network agreed upon by the participants.
Indexed ? :	The Coordinating Centre may index this field using Indexing Technique 0.

Field Tag:	002
Field Name:	Participating centre record number
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None
Representation:	Accession number, mfn or other record number assigned by the inputting centre
Indexed ? :	N

Field Tag:	003
Field Name:	Record status
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Representation:	Alphabetical. N for new record; R for revised record; D for deleted record
Indexed ? :	The Coordinating Centre may index this field using Indexing Technique 0.

Field Tag:	005
Field Name:	Date record entered
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	Record is entered on 28 February 1989. Enter 1989-02-28
Indexed ? :	N
Remarks:	This implementation has been set up to accept a date in the format 9999-99-99, e.g. 1989-08-23 for 2 August 1989.

Field Tag:	006
Field Name:	Date record changed
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY.
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	Record is changed on 28 March 1989. Enter 1989-03-28
Indexed ? :	N
Remarks:	This implementation has been set up to accept a date in the format 9999-99-99 e.g. 1989-08-23 for 2 August 1989.

Field Tag:	007
Field Name:	Bibliographic level
Applicable to:	Bibliographic materials only (those having codes in field 420)
Worksheet:	ABNCD. PARTM. PARTS.
Repeatable ? :	N
Subfields:	None; these have not been used in this implementation
Representation:	Refer MIBIS.
Example :	Other than a blank, 'a', 'm', 's' and 'c' are the only characters permitted in this field.
Indexed ? :	N
Remarks:	This field has not been inverted though for certain applications it may be essential to do so.

Field Tag: 005
Field Name: Date record entered
Applicable to: All types of records
Worksheet: ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? : N
Subfields: None.
Representation: Refer MIBIS.
Example : Record is entered on 28 February 1989. Enter 1989-02-28
Indexed ? : N
Remarks: This implementation has been set up to accept a date in the format 9999-99-99, e.g. 1989-08-23 for 2 August 1989.

Field Tag: 006
Field Name: Date record changed
Applicable to: All types of records
Worksheet: ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY.
Repeatable ? : N
Subfields: None.
Representation: Refer MIBIS.
Example : Record is changed on 28 March 1989. Enter 1989-03-28
Indexed ? : N
Remarks: This implementation has been set up to accept a date in the format 9999-99-99 e.g. 1989-08-23 for 2 August 1989.

Field Tag: 007
Field Name: Bibliographic level
Applicable to: Bibliographic materials only (those having codes in field 420)
Worksheet: ABNCD. PARTM. PARTS.
Repeatable ? : N
Subfields: None; these have not been used in this implementation
Representation: Refer MIBIS.
Example : Other than a blank, 'a', 'm', 's' and 'c' are the only characters permitted in this field.
Indexed ? : N
Remarks: This field has not been inverted though for certain applications it may be essential to do so.

Field Tag: 008
Field Name: Bibliographic level - parent
Applicable to: Bibliographic materials only (those having codes in field 420)
Worksheet: ABNCD. PARTM. PARTS.
Repeatable ? : N
Subfields: None; these have not been used in this implementation
Representation: Refer MIBIS.
Example : Other than a blank, 'a', 'm', 's' and 'c' are the only characters permitted in this field.
Indexed ? : N
Remarks: This field has not been inverted though for certain applications it may be essential to do so.

Field Tag: 009
Field Name: Country of origin.
Applicable to: All types of records.
Worksheet: ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? : N
Subfields: None.
Representation: Two letter ISO code for country which inputs the record. Refer MIBIS Annex 10.
Example : Record is input by India. Enter IN
Indexed ? : The Coordinating Centre may index this field using Indexing Technique 0.

Field Tag: 010
Field Name: Record number of parent
Applicable to: Bibliographical records - analytics.
Worksheet: PARTM. PARTS
Repeatable ? : N
Subfields: None.
Representation: Numerical. MFN of parent document if catalogued in the same database.
Indexed ? : N

Field Tag: 011
Field Name: Record number of parts
Applicable to: Bibliographical record - monograph
Worksheet: ABNCD
Repeatable ? : Y
Subfields: None
Representation: Numerical. MFNs of records of parts of the document being entered if the parts are catalogued in the same database.
Indexed ? : N

Field Tag:	012
Field Name:	Record number of other language version(s).
Applicable to:	Bibliographical records
Worksheet:	ABNCD
Repeatable ? :	Y
Subfields:	None.
Representation:	Numerical. MFNs of records of other language versions of the document catalgued if they are entered in the same database.
Indexed ? :	N

Field Tag:	020
Field Name:	Language of analysis
Applicable to:	All types of records
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Enter two-digit language code. Refer MIBIS Annex 8.
Example :	The record is entered in Spanish. Enter sp
Indexed ? :	N (optional)
Remarks:	This field is of use when records are entered in more than one language as in a database in a multi-lingual organization. It will be necessary to add the field to each Worksheet. If it is desired to add the field to each record for transfer to another database, it should be added via the FST used as the conversion parameter file.

Field Tag:	021
Field Name:	Language(s) of text.
Applicable to:	Bibliographical record.
Worksheet:	ABNCD. PARTM. PARTS
Repeatable ? :	Y
Subfields:	None.
Representation:	See Field Tag 020
Example :	The document is in English. Enter: en. The document is in French and Spanish. Enter: fr%sp
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

Field Tag:	022
Field Name:	Language of summaries.
Applicable to:	Bibliographical record and project record.
Worksheet:	ABNCD. PARTM. PARTS. ABNRP
Repeatable ? :	Y
Subfields:	None.
Representation:	Use 2-letter language codes. Refer MIBIS Annex 8.
Example :	A serial includes summaries of each article in French, English and Spanish. Enter en%fr%sp
Indexed ? :	Optional.

Field Tag:	100
Field Name:	Title
Applicable to: system	Bibliographic records. Project. Information
Worksheet:	ABNCD. PARTM. PARTS. ABNRP. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation: Similarly	Refer MIBIS for bibliographic materials. enter name (title) of project and name of information system.
Example :	Title of document given on the title page is Asterix in Switzerland. Enter: Asterix in Switzerland. For system known as BLAISE, enter: BLAISE For project titled 'An investigation of late Quaternary continental shelf sediments, enter: An investigation of late Quarternary continental shelf sediments.
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole

GUIDE TO DATA ENTRY IN ABNCD DATABASE

Field Tag:	101
Field Name:	Parallel title(s)
Applicable to:	Bibliographic record. Project. Information system.
Worksheet:	ABNCD. PARTM. PARTS. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Refer MIBIS for bibliographic record. Similarly enter parallel titles, if any, for a research propject and an information system.
Example :	Title proper is: British standard methods of analysis of fat and fatty oil. Enter parallel titles: Methodes d'analyse des graisses et huiles fixes%Untersuchungsverfahren fur Fette und Fettole
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole

Field Tag:	102
Field Name:	Translated title - English
Applicable to:	Bibliographic record. Project, Information System.
Worksheet:	ABNCD. PARTM. PARTS. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Refer MIBIS for bibliographic record. Similarly for project and information system if applicable.
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole.

Field Tag:	105
Field Name:	Translated title - other
Applicable to:	Bibliographic record. Project. Information system.
Worksheet:	ABNCD. PARTM. PARTS. ABNRP. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS for bibliographic record. Similarly for propject and information system if applicable.
Indexed ? :	N

Field Tag:	110
Field Name:	Name of person
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTS. PARTM. ABNIN. ABNRP. ABNSY. ABNEX
Repeatable ? :	Y
Subfields:	A, Name of person. B, Role
Representation:	Refer MIBIS.
Example :	Abebe Rorissa and A. Neelameghan are authors of a paper in a monograph. Enter: ; Abebe Rorissa%; Neelameghan, A. R.B. Stokes is the editor of a monograph. Enter: ; Stokes, R. B., ed. Dean Jones is the director of an institution. Enter: ; Jones, Dean, Director. The profile of Anup Agarwal is being recorded. Enter: ; Agarwal, Anup John Gregorio is coordinator of a project. Enter: ; Gregorio, John, Coordinator
Indexed ? :	Y
Index Technique:	0
Whole or SB:	
Remarks:	The role may be indexed into a different field so that searches can be done for editors, directors, project managers, etc. (See also 210 for personal author relating to a higher level (e.g. author of monograph when a record is that of an analytic in a monograph).

Field Tag:	111
Field Name:	Name of corporate body
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	a, Main body. b, Sub-body. c, Place. d, Country
Representation:	Refer MIBIS.
Example :	For Library of the Punjab University, enter: ; Punjab University, Library, Ludhiana, IN For Government of Sri Lanka, Ministry of Social Welfare, enter: ; Sri Lanka, Ministry of Social Welfare, Colombo, LK
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Subfields a, b
Remarks:	Database managers may find indexing technique 4 more suitable so that corporate body words will be inverted individually. (See also 211 for corporate body relating to a higher level (e.g. author of monograph when the record is that of an analytic in a monograph).

Field Tag:	112
Field Name:	Affiliation
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNEX. ABNIN. ABNSY
Repeatable ? :	Y
Subfields:	a, Main body. b, Sub-body. c, Place. d, Country
Representation:	Refer MIBIS.
Example :	A person is affiliated to the Department of Southeast Asian Studies of the University of the Philippines. Enter: ; University of the Philippines, Department of Southeast Asian Studies, Manila, PH
Indexed ? :	Optional
Remarks:	Mention of affiliation is optional. If entered the corporate names in Subfields a and b may be inverted if it was felt desirable; see field 111. Depending on the cataloguing rules, affiliations that are important enough to be inverted could appear as a corporate body in field 111.

Field Tag:	113
Field Name:	Other associated institutions.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNEX. ABNRP. ABNSY.
Repeatable ? :	Y
Subfields:	a, Main body. b, Sub-body. c, Place. d, Country code. e, Role
Representation:	Refer MIBIS.
Example :	The International Development Research Centre in Ottawa, as a funding agency, enter: ; International Development Research Centre , Ottawa, CA, Funder
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Subfields a and b.
Remarks:	One may use indexing technique 4.

Field Tag:	114
Field Name:	Meeting.
Applicable to:	Bibliographic record.
Worksheet:	ABNCD. PARTS. PARTM.
Repeatable ? :	Y
Subfields:	a, Meeting name and number. b, Place. c, Country code. d, Dates. e, Date in ISO form.
Representation:	Refer MIBIS.
Example :	A conference takes place in New York throught 1973. Enter: ; United Nations Conference on the Law of the Sea, 3rd, New York, US, 1973, 19730000. If complete dates of the meeting are known enter: ; International Symposium on Bibliographic Exchange Formats, Taormina, IT, 1978-04-26/ 28, 19780426-19780428.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Subfield a
Remarks:	Alternatively indexing technique 4 may be used.

Field Tag:	115
Field Name:	Translated name of corporate body
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	a, Main body. b, Sub-body. c, Place. d, Country
Representation:	Refer MIBIS.
Example :	In information systems where the name of the corporate body must be in one of the languages of the system, field 115 is used when the name of the corporate body entered in field 111 is in a language other than the language of analysis.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Subfields a and b
Remarks:	In information systems where the name of the corporate body must be in one of the languages of the system, field 115 is used when the name of the corporate body entered in field 111 is in a language other than the language of analysis. An alternative indexing technique is 4.

Field Tag:	116
Field Name:	Address/Phone, etc.
Applicable to:	Institution. Project. System. Person.
Worksheet:	ABNIN. ABNRP. ABNSY. ABNEX
Repeatable ? :	N
Subfields:	a, Street address. b, P.O. Box. c, City/Town + ZIP/PIN code. d. District/State. e, Country. f, Phone. g, Cable. h, Teepresentation..: Refer MIBIS.
Example :	257 p, ill., 4 slides, 23 cm ; 350 p., graphs, 120 ref. ; 320 p., bibliog. p. 310-316 ; 1 v. in various paginations. ; 2 v. (loose-leaf) ; 12 maps ; 1 audiocassette (120 min.) ; 1 videocassette (30 min.)
Indexed ? :	N
Remarks:	Mention of dimension is optional except in the case documents that are extra large or abnormally small.

Field Tag:	131
Field Name:	Part statement
Applicable to:	Bibliographic record.
Worksheet:	PARTM. PARTS
Repeatable ? :	Y
Subfields:	a, Volume/part/issue number. b, Pagination of part.
Representation:	Refer MIBIS.
Example :	Vol. 36, 25-36. ; v. 14, no. 5. ; no. 25
Indexed ? :	N
Remarks:	Use 130 for pagination of monographs.

Field Tag:	140
Field Name:	Series
Applicable to:	Bibliographic record. Project.
Worksheet:	ABNCD. PARTS. ABNRP
Repeatable ? :	Y
Subfields:	a, Series title. b, Part no.
Representation:	Refer MIBIS.
Example :	Occasional paper, British Museum, no. 33 ; Food research projects
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole

Field Tag:	141
Field Name:	Thesis.
Applicable to:	Bibliographic record. Project.
Worksheet:	ABNCD. PARTM. PARTS. ABNRP
Repeatable ? :	N
Subfields:	a, Thesis designation. b, Degree. c, Course. d, Number.
Representation:	Refer MIBIS.
Example :	Thesis, Ph. D., Law, no. 33 ; Dissertation, M. S., Medical Sciences
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole

Field Tag:	142
Field Name:	Related project(s)
Applicable to:	Bibliographic record. Project.
Worksheet:	ABNCD. PARTM. PARTS. ABNRP
Repeatable ? :	Y
Subfields:	a, Project title. b, Project no.
Representation:	Refer MIBIS.
Example :	River Basin Development, PH-83-10 ; Low-cost Housing Project, IN-84-34
Indexed ? :	Y
Index Technique:	4
Whole or SB:	Whole

Field Tag:	150
Field Name:	Notes
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNSY.
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	The result of a cooperative research project at the Federal Highway Research Institute, Bergisch Gladbach, Federal Republic of Germany
Indexed ? :	N
Remarks:	Optional.

Field Tag:	160
Field Name:	ISBN
Applicable to:	Bibliographical record - monograph.
Worksheet:	ABNCD
Repeatable ? :	Y
Subfields:	one
Representation:	Refer MIBIS.
Example :	0-7214-0191-0. 0-8389-3211-8 pbk
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

Field Tag:	161
Field Name:	Document number
Applicable to:	Bibliographic record.
Worksheet:	ABNCD. PARTM
Repeatable ? :	Y
Subfields:	None.
Representation:	Refer MIBIS.
Example :	IDRC-TS67e ST/ESA/SER.M/11
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

Field Tag:	162
Field Name:	Availability
Applicable to:	Bibliographic record.
Worksheet:	ABNCD
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Enter information on where the document can be obtained if it is not available from the the issuing body. Indicate whether the document is restricted in cirulation/distribution, confidential etc.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

GUIDE TO DATA ENTRY IN ABNCD DATABASE

Field Tag:	200
Field Name:	Title of serial
Applicable to:	Bibliographic record.
Worksheet:	PARTS
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	Journal of Development Studies IDRC reports Economic report, Asian Development Bank
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Remarks:	An alternative is to use indexing technique 4.

Field Tag:	201
Field Name:	ISSN (International Standard Serial Number)
Applicable to:	Bibliographical record - Serial
Worksheet:	ABCD. PARTS
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. According to the provisions of ISO 3297. Pattern: 9999-999X
Example :	The ISSN on a serial is ISSN 0262-7264. Enter: 0262-7264
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

Field Tag:	202
Field Name:	Title of parent (M/C)
Applicable to:	Bibliographic record. (Multi-level bibliographic material)
Worksheet:	PARTM
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. When item catalogued is a chapter or a section or a part of a monograph or a collection (with bibliographic level AM, AC or MC) enter the title of the parent item in field 202.
Example :	The item catalogued has the title "Rural energy systems in Indonesia" and is a chapter in the monograph entitled "Integrated rural energy development." Enter in field 100: Rural energy systems in Indonesia. Enter in field 202: Integrated rural energy development.
Indexed ? :	N
Remarks:	This field is not inverted as it is purely the location of the item.

Field Tag:	210
Field Name:	Personal author(s) - parent.
Applicable to:	Bibliographic record - analytic.
Worksheet:	PARTM. PARTS
Repeatable ? :	Y
Subfields:	a, Name of author. b, Role.
Representation:	Refer MIBIS. Item catalogued is a chapter or a contribution from a monograph, enter in field 210 the name of author, compiler or or editor as the case may be, of the parent item.
Example :	The item catalogued is a paper from the proceedings of a conference edited by Frank Smith. Enter: ; Smith, Frank, ed.
Indexed ? :	N
Remarks:	This field is not inverted as it relates purely to the document in which the item being recorded is found.

Field Tag:	211
Field Name:	Corporate author(s) - parent
Applicable to:	Bibliographic record - analytic
Worksheet:	PARTM. PARTS
Repeatable ? :	Y
Subfields:	a, Main body. b, Sub-body. c, Place. d. Country code.
Representation:	Refer MIBIS. Item catalogued is a chapter or a contribution from a monograph, enter in field 211 the name of the corporate author(s) of the parent item.
Example :	A paper being catalogued is authored by Jean Drummond and published in the volume of proceedings of a conference edited by the Institute of Information Studies in Bangalore, India. Enter in field 211 ; Institute of Information Studies , Bangalore, IN
Indexed ? :	N
Remarks:	This field is not inverted (cf 210).

Field Tag:	300
Field Name:	Primary descriptors.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTS. PARTM. ABNVPP. ABNIN. ABNRP. ABNEX. ABNSY.
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	The main subject(s) dealt with in a document or the field of specialization of an insitution, of an expert, or of a project, or of an information system is Pest Control of Cotton Plants, then the descriptors to be entered in field 300 could be: Agriculture. Cotton. Pests. Control. For using indexing technique 2, enter: AGRICULTURE COTTON PESTS CONTROL
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.
Remarks:	Enter terms denoting geographical area in field 302.

Field Tag:	301
Field Name:	Secondary descriptors.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTS. PARTM. ABNVPP. ABNIN. ABNRP. ABNEX. ABNSY.
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Enter in field 301 subject descriptors selected from a thesaurus or other vocabulary control tool that were not entered in field 300.
Example :	SOCIAL ENVIRONMENT CASE STUDIES
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.
Remarks:	Enter terms denoting geographical area in field 302.

Field Tag:	302
Field Name:	Geographical descriptors.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	The subject(s) dealt with in a document or the field of specialization of an insitution, of an expert, or of a project, or of an information system is Agriculture in Asia with special reference to the Philippines. The descriptor to be entered in field 300 will be: Agriculture, and the descriptors to be entered in field 301 will be: Asia. Philippines. For using indexing technique 2, enter: ASIA PHILIPPINES
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.

Field Tag:	303
Field Name:	Local descriptors.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Enter descriptors that the system requires and not entered in fields 300, 301, or 302 and not found in the thesaurus used.
Example :	The subject(s) dealt with in a document or the field of specialization of an insitution, of an expert, or of a project, or of an information system is Women in India. The descriptor to be entered in field 300 will be: Women, and in field 302 the descriptor India. In field 303 may be entered SATI PRACTICE ROY, RAJA RAM MOHUN VIDYASAGAR, ISWAR CHUNDER
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.

Field Tag:	304
Field Name:	Proposed descriptors.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS.
Example :	The subject(s) dealt with in a document or the field of specialization of an insitution, of an expert, or of a project, or of an information system is Women in Development. The descriptor to be entered in field 300 will be: Women. The information centre may wish to propose for consideration of the participants of the network the inclusion of the term Sati Practice as a descriptor in the thesaurus or other vocabulary control tool used. Enterin field 304: SATI PRACTICE
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.

Field Tag:	305
Field Name:	IRC subject classification.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Subject class number taken from the IRC scheme.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.
Remarks:	Enter only one class number for a record.

Field Tag:	310
Field Name:	Abstract/Description
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	According to practices of agency preparing the abstract or description of an entity. Words to be indexed should be enclosed in angular brackets ... for using indexing technique 2 Alternatively, indexing technique With a stopword file can also be used.
Example :	Preliminary investigation of a microcomputer as an aid in both local and network information handling indicates that there are major benefits. The system has proven useful in instructional settings and in approximations of normal library-information centre tasks.
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.

Field Tag:	320
Field Name:	Broad subject heading.
Applicable to:	All types of records.
Worksheet:	ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. The Broad Subject Heading entered in field 320 can be used to arrange entries in a current awareness bulletin generated from the database.
Example :	Child care
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole.
Remarks:	Enter only one broad subject heading for a record.

Field Tag:	400
Field Name:	Processing status
Applicable to:	Bibliographic records.
Worksheet:	ABNCD
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Indicate by an appropriate code the stage of processing of the document e.g. ORD (item is on on order); CAT (item has been catalogued), etc.
Indexed ? :	N
Remarks:	The notation in the field should be changed as the work goes through different stages of processing. Alternatively, the notation for each stage of processing can be added one after the other e.g. ORD. RECD. CAT...

Field Tag:	410
Field Name:	Location.
Applicable to:	Bibliographic record.
Worksheet:	ABNCD, PARTM. PARTS
Repeatable ? :	Y
Subfields:	None.
Representation:	Refer MIBIS. Indicate in which library/library collection the item is kept. Field 4Will indicate the exact location of the item by the call number.
Indexed ? :	N

Field Tag:	411
Field Name:	Call number.
Applicable to:	Bibliographic record.
Worksheet:	ABNCD
Repeatable ? :	Y
Subfields:	None.
Representation:	According to provisions of the classification scheme used.
Example :	681.327.54'11 UNI. 616.122 UNI
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole

Field Tag: 412
Field Name: Number of copies.
Applicable to: Bibliographic record.
Worksheet: ABNCD
Repeatable ? : N
Subfields: None.
Representation: Refer MIBIS.
Example : The library has 3 copies. Enter: 3.
Indexed ? : N

Field Tag: 415
Field Name: Accession number
Applicable to: Bibliographic record.
Worksheet: ABNCD
Repeatable ? : N
Subfields: None
Representation: According to local practice
Example : 32456
Indexed ? : N
Remarks: Optional. Accession number is usually given in the local/institutional catalogue entry.

Field Tag: 420
Field Name: Type of material
Applicable to: Bibliographic record.
Worksheet: ABNCD. PARTM. PARTS
Repeatable ? : N
Subfields: None.
Representation: Refer MIBIS. Use codes for Type of Material.
Example : PER for periodical article. MONO for monograph. PART for part of a monograph or collection. THES for thesis. REP for report. COD for conference document. UND for United Nations document.
Indexed ? : Y
Index Technique: 0
Whole or SB: Whole.

Field Tag: 430
Field Name: Documentalist.
Applicable to: All types of records.
Worksheet: ABNCD. PARTM. PARTS. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? : Y
Subfields: None.
Representation: Refer MIBIS. Enter the name(s) or initials of the person(s) entering the record.
Indexed ? : N

Field Tag:	441
Field Name:	Duration.
Applicable to:	Project.
Worksheet:	ABNRP
Repeatable ? :	N
Subfields:	None.
Representation:	Enter the duration of the project.
Example :	1986-1988. 1988-05 to 1989-07. 6 months.
Indexed ? :	N

Field Tag:	442
Field Name:	Date of proposal/approval
Applicable to:	Bibliographic record - acquisitions. Projects. Systems.
Worksheet:	ABNCD. ABNRP. ABNSY
Repeatable ? :	N
Subfields:	a, Date of proposal. b, Date of approval
Representation:	Date of proposal - format:YYYYMMDD. Date of approval - format:YYYYMMDD
Example :	Dates of proposal and approval for a project are respectively 9 June 1988 and 12 July 1988.
Enter:	; 19880609, 19880712
Indexed ? :	N
Remarks:	Information is usually of local/institutional interest. If the data base is used for Documents Acquisitions management, the field can be used for indicating the date of receipt of proposal and date of approval of the proposal for acquisition of a document.

Field Tag:	443
Field Name:	Date of starting
Applicable to:	Project. System.
Worksheet:	ABNRP. ABNSY
Repeatable ? :	N
Subfields:	None
Representation:	Format:YYYYMMDD
Example :	Date of starting of a project or system is 5 August 1989. Enter 19890805
Indexed ? :	N

Field Tag:	444
Field Name:	Date of expected completion
Applicable to:	Project.
Worksheet:	ABNRP
Repeatable ? :	N
Subfields:	None
Representation:	Format:YYYYMMDD
Example :	Date of expected completion of a project is 12 May 1990. Enter 19900512
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Remarks:	The 'date of expected completion' is usually of local/institutional interest.

Field Tag:	445
Field Name:	Date of actual completion
Applicable to:	Project.
Worksheet:	ABNRP.
Repeatable ? :	N
Subfields:	None
Representation:	Format:YYYYMMDD
Example :	Date of actual completion of a project is 15 Janaury 1988. Enter 19880115
Indexed ? :	N

Field Tag:	446
Field Name:	Date terminated
Applicable to:	Project. System.
Worksheet:	ABNRP. ABNSY.
Repeatable ? :	N
Subfields:	None
Representation:	Format:YYYYMMDD
Example :	The date of termination of a project or system is 4 October 1989. Enter: 19891004
Indexed ? :	N

Field Tag:	447
Field Name:	Date of birth
Applicable to:	Person
Worksheet:	ABNEX
Repeatable ? :	N
Subfields:	None.
Representation:	YYYYMMDD.
Example :	Enter: 1945121For 12 December 1945. 1947 for 1947. 194805 for May 1948.
Indexed ? :	N

Field Tag:	500
Field Name:	Acquisition type.
Applicable to:	Bibliographic record - acquisitions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	None
Representation:	Refer MIBIS for field 500 and Annex 6.
Example :	Enter: PUR for purchased item. SUB for subsribed item. FREE for gifts, or acquired free. EXCH for items acquired on exchange.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Remarks:	The field is inverted to enable retrieval of of all purchased materials, subscribed items, those obtained free, etc.

Field Tag:	501
Field Name:	Date ordered.
Applicable to:	Bibliographic record - acquisitions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Format YYYY-MM-DD.
Example :	Enter: 1991-12-1For 12 December 1991.
Indexed ? :	N

Field Tag:	511
Field Name:	Date claimed.
Applicable to:	Bibliographic record - acquisitions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Format YYYY-MM-DD.
Example :	Enter: 1992-05-1For 12 May 1992.
Indexed ? :	N

Field Tag:	512
Field Name:	Date received.
Applicable to:	Bibliographic record - acquisitions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	None.
Representation:	Refer MIBIS. Format YYYY-MM-DD.
Example :	Enter: 1992-03-05 for 5 March 1992.
Indexed ? :	N

Field Tag: 513
Field Name: Number of copies ordered.
Applicable to: Bibliographic record - acquisitions.
Worksheet: ACQIS
Repeatable ? : N
Subfields: None.
Representation: Refer MIBIS.
Example : For three copies ordered, enter: 3194805 for May 1948.
Indexed ? : N

Field Tag: 514
Field Name: Requester or proposer.
Applicable to: Bibliographic record - acquisitions and project.
Worksheet: ACQIS. ABNRP
Repeatable ? : Y
Subfields: None.
Representation: Refer MIBIS. Enter the name(s) of person(s) requesting the procurement of the document or submitting a research proposal.
Example : Juanito gregorio%Tatiana Kodyat.
Indexed ? : N

Field Tag: 515
Field Name: Supplier.
Applicable to: Bibliographic record - acquisitions.
Worksheet: ACQIS
Repeatable ? : N
Subfields: a, Supplier name. b, Address - line 1. c, Address - line 2. d, Address - line 3. e, country.
Representation: Refer MIBIS. Enter the name and address of the supplier of of the item.
Example : India Book House, Gandhibazaar , Bangalore-560009, India.
Indexed ? : N
Remarks: An alternative is to give only a subfield e.g. for entering a supplier code, create a supplier authority file, and use the REF function (in the display format) to establish link with the appropriate supplier record.

Field Tag:	516
Field Name:	Price.
Applicable to:	Bibliographic record - acquisitions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	A, Price. B, Currency.
Representation:	Refer MIBIS. Use ISO 4217 Currency Code.
Example :	In the country of publication, the paperback edition sold for $10.35. Enter: ; 10.35, USD
Indexed ? :	N

Field Tag:	517
Field Name:	Acquisition notes
Applicable to:	Bibliographic record - acquistions.
Worksheet:	ACQIS
Repeatable ? :	N
Subfields:	None.
Representation:	Notes such as, Standing order, Rush order...; that the document received was returned on (date) because it was an earlier edition, incorrect collation...; that the order was cancelled and placed with...., etc. Note of messages received from the supplier e.g. Out of stock; out of print, etc.
Indexed ? :	N

Field Tag:	525
Field Name:	Language competence/workind languages.
Applicable to:	Expert. Institution. Project. System.
Worksheet:	ABNEX. ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	a, Language. b. Competence in. c, Level of competence.
Representation:	Language: ISO 2-letter language code (Refer MIBIS Annex 8). Competence in: Read, Write, Speak. Level of excellence: Fair, Good, Excellent or abbreviations such as F, G, E respectively. Example : For person: ; En, Read, Write, Speak, Excellent%;Fr , Read, Speak, Fair For Institution: ; En%; Fr
Indexed ? :	Y
Index Technique:	0
Whole or SB:	

Field Tag:	556
Field Name:	Assignments.
Applicable to:	Expert.
Worksheet:	ABNEX
Repeatable ? :	Y
Subfields:	a, Place/Country. b. Institution. c. Purpose. d, Period
Representation:	Country: Code for country assigned/visited. Instiution: Name and place of institution visited. Purpose: e.g study, consultancy, etc Period: Inclusive dates if possible.
Example :	GB, Institute of Education, London, Study visit, June-July 1987
Indexed ? :	N
Remarks:	Local/institutional interest.

Field Tag:	570
Field Name:	Personnel
Applicable to:	Institution. Project. System.
Worksheet:	ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	a, Category of personnel. b, Number
Representation:	Category: e.g. Technical, Administrative, etc. Number: No. of persons in the category.
Example :	Technical, 25%; Administrative, 12
Indexed ? :	N

Field Tag:	625
Field Name:	Objectives.
Applicable to:	Institution. Project. System.
Worksheet:	ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	Text prepared from the constitution, charter, terms of reference or other source for the entity concerned.
Example :	For an organization: To conduct training courses in water and sanitation to extension workerss in rural areas; and to provide financial assistance for the conduct of such courses.
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole

Field Tag:	700
Field Name:	Financial aspects
Applicable to:	Institution. Project.
Worksheet:	ABNIN. ABNRP
Repeatable ? :	Y
Subfields:	s, Source. a, Amount. c, Currency. p, Purpose.
Representation:	Source: Source of finance e.g. IDRC; own... Purpose: Purpose for which the amount has been given e.g. Equipment.
Example :	DANIDA; 12,000, DG, Equipment.
Indexed ? :	Y
Whole or SB:	

Field Tag:	830
Field Name:	Nationality/Coutry of registration.
Applicable to:	Institution. Person. Project.
Worksheet:	ABNIN. ABNEX. ABNRP
Repeatable ? :	Y
Subfields:	None.
Representation:	Enter the country code (MIBIS Annex 10) of the nationality of the person or country of registration of the institution or project.
Example :	IN%LN%PH
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	831
Field Name:	Qualifications.
Applicable to:	Person.
Worksheet:	ABNEX
Repeatable ? :	Y
Subfields:	a, Degree/Diploma. b, Institution. c, Major subject. d, Date of award.
Representation:	Degree/Dip.: Use standard abrreviations. Institution: Name of institution awarding the degree... Major subject: e.g. Chemistry. Date: Year of award.
Example :	M.Sc, Univ. Philippines, Botany. 1982% ; Ph.D, Texas A&M, Agrifulture, 1986
Indexed ? :	Y
Index Technique:	0
Whole or SB:	

Field Tag:	833
Field Name:	Work experience
Applicable to:	Person
Worksheet:	ABNEX
Repeatable ? :	Y
Subfields:	a, Designation. b, Environment. c, Institution, Place. d, Period
Representation:	Designation: Name of of the post. Evironment: Type of field of work (e.g. Industry, Research laboratory). Institution, Place: Name of institution where the person was employed. Period: Dates in the post, in free form
Example :	Programmer, University, Univ. of Delhi, 1982 Jan.-198Aug%; System Analyst, University , IIT, Delhi, 1983 Sep.-1985 June% ; Head, Government, Computer Centre, NSO, Delhi, 1985 July-
Indexed ? :	Y
Index Technique:	0
Whole or SB:	
Remarks:	The sequence of enumeration of posts may be the current one first and going backward in time.

Field Tag:	834
Field Name:	Current work
Applicable to:	Person
Worksheet:	ABNEX
Repeatable ? :	N
Subfields:	a, Designation. b, Employer. c, Date
Representation:	Designation: Name of the current post held. Employer: Name of institution where employed. Date: Date of joining the present post
Example :	Principal Scientific Officer, National Res. Council, Jakarta, 1989 Aug.-
Indexed ? :	N
Remarks:	This field is optional. The information can be accommodated in field 833 Work experience. However, this field can be used when in the printed output information is to be provided under a separate heading 'current work'.

Field Tag:	835
Field Name:	Marital status
Applicable to:	Person
Worksheet:	ABNEX
Repeatable ? :	Y
Subfields:	None.
Representation:	Marital status: Indications such as, Married, Divorced, Widow, etc.
Example :	Separated.
Indexed ? :	N
Remarks:	May be of Local/institutional interest only.

Field Tag:	836
Field Name:	Sex
Applicable to:	Person.
Worksheet:	ABNEX
Repeatable ? :	N
Subfields:	None.
Representation:	Male or Female.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Field Tag:	850
Field Name:	Recommended by
Applicable to:	Document. Institution. Project. Person. System. Activity
Worksheet:	ABNCD. ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	a, Name. b. Designation/Affiliation. c, Recommended for. d, Date
Representation:	Name: Name of person recommending the entity, rendered Surname, Forename(s). Designation/Institution: Name of position held and/or affiliation of the recommending person. Recommended for: What the entity has been recommended for. Date: Date of recommendation in free form
Example :	A. Neelameghan recommended G. Bhattacharyya for a Consultant mission. In the profile record of Bhattacharyya the following entry may be made: ; Neelameghan, A, DRTC/ISI, Bangalore, Consultancy for Unesco, 1986
Indexed ? :	Y
Index Technique:	0
Whole or SB:	
Remarks:	The information may be of local/institutional interest only.
Field Tag:	855
Field Name:	Honours and awards
Applicable to:	Institution. Project. Person. System.
Worksheet:	ABNIN. ABNRP. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	a, Name of award. b, Awarding body. c, Date
Representation:	Name of award: Name of the honour conferred. Awarding body: Name of body conferring the honour or award. Date: Date of award in free form
Example :	D.Lit.(Honoris causa), Univ. of Delhi, 1946% ; Padmasri, Govt. of India, 1967
Indexed ? :	N
Remarks:	The information may be of local/institutional interest only

Field Tag:	856
Field Name:	Membership in associations
Applicable to:	Institution. Person. System
Worksheet:	ABNIN. ABNEX. ABNSY
Repeatable ? :	Y
Subfields:	a, Name of association. b. Type of membership. c, Year. d, Special position/office held
Representation:	Name of association: Name of association in abbreviated form. Type of membership: e.g. ordinary member, institutional member, life member, honorary member. Year: Year/period of membership. Special position/office held: Special position or office held in the association, e.g. Treasurer
Example :	Brit. Assoc. Adv. Sci, Member, 1975-% ; Chemical Soc., London, Life member, 1976- , Secretary, 1981-1984
Indexed ? :	N
Remarks:	This information may be of local/Institutional interest only

Field Tag:	890
Field Name:	Patents taken
Applicable to:	Institution. System.
Worksheet:	ABNIN. ABNSY
Repeatable ? :	Y
Subfields:	a, Patent number. b, Title. c, Inventor. d. Assignee. e, Assignor. f, Country. g, Date. h, Coverage
Representation:	Patent number: Format:US 723456. Title: Title of patent/application. Inventor: Name of inventor, rendered Surname, Forename(s). Assignee: Name of body or person to whom the patent is assigned by the inventor. Assignor: Name of person or body, if different from the inventor, assigning the patent. Country: Name of country where the patent is originally registered. Date: Date of application/registration of patent. Coverage: Countries where the patent has concurrent validity
Example :	Jap 78/3456, A method for purifying pentaenes, Matsuoka, I, Tokyo Pharmaceutical Co., Antibiotics Research Lab., Kyoto, Japan , 1978 May, USA
Indexed ? :	Y
Index Technique:	1
Whole or SB:	All Subfields
Remarks:	Bibliographical details for a patent document is to be entered in the ABNCD Worksheet.

Field Tag:	896
Field Name:	Classification system used.
Applicable to:	Institution. Project. System.
Worksheet:	ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Abbreviated standard name of the scheme.
Example :	UDC%IRC classification.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	897
Field Name:	Subject headings list used.
Applicable to:	Project. System.
Worksheet:	ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Abbreviated standard name of the list used.
Example :	LC subject headings list%Sears list of subject headings.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	898
Field Name:	Thesaurus used.
Applicable to:	Institution. Project. System.
Worksheet:	ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Abbreviated standard name of the thesaurus.
Example :	Macrothesaurus%AGROVOC
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	899
Field Name:	Periodical publications.
Applicable to:	Institution. Project. System.
Worksheet:	ABNIN. ABNRP. ABNSY
Repeatable ? :	Y
Subfields:	None.
Representation:	Title of the publication.
Example :	ISI bulletin%Sankhya
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	900
Field Name:	Products and services
Applicable to:	Institution. Project. System. Person.
Worksheet:	ABNIN. ABNRP. ABNSY. ABNEX
Repeatable ? :	Y
Subfields:	None.
Representation:	In free form
Example :	Data bases: CDS - bibliographic, 1978-; FORD - forest resources, 1989-. Current awareness service. SDI
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Terms within angle brackets.

Field Tag:	803
Field Name:	Name of System/Equipment
Applicable to:	System. Activity
Worksheet:	ABNSY. DIVAC
Repeatable ? :	Y
Subfields:	A, System/Equipment
Representation:	Use trade names or other unique identifiers
Example :	AUTODOC%; MINISIS%; CDS/ISIS
Indexed ? :	Y
Index Technique:	0
Whole or SB:	whole

Field Tag:	804
Field Name:	Note on data collection
Applicable to:	Systems and services
Worksheet:	ABNSY
Repeatable ? :	N
Subfields:	A, Note on data collection
Representation:	In free form
Example :	Data are collected through questionnaires sent out at annual intervals
Indexed ? :	N

Field Tag:	805
Field Name:	Number of records
Applicable to:	Systems and services
Worksheet:	ABNSY
Repeatable ? :	Y
Subfields:	A, No. of records
Representation:	In free form
Example :	No. of records 5000%Annual intake 1000
Indexed ? :	N

Field Tag:	810
Field Name:	Sources scanned
Applicable to:	Document (Periodical publication). System
Worksheet:	DIVPP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	For an abstracting/indexing serial, database, information base indicate the approximate number of books, serials, etc. covered each year or in each volume
Example :	Covers approx. 2400 serial titles, reviews 250 books; indexes over 2000 reports each year
Indexed ? :	N
Remarks:	Optional
Field Tag:	812
Field Name:	Contents and arrangement
Applicable to:	Document
Worksheet:	ABNCD. DIVPP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	Especially for an abstracting/indexing serial, directory/inventory, or data base indicate how the entries are organized and arranged
Example :	In an abstracting periodical in the main part: Entries are arranged by subject categories and under each subject category by name(s) of author(s) in alphabetical sequence. A list of the subject categories used is given in page xi of each issue
Indexed ? :	N
Remarks:	Optional
Field Tag:	814
Field Name:	Typical entry
Applicable to:	Document. System
Worksheet:	ABNCD. DIVPP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	Especially for an abstracting/indexing serial, directory/inventory, or data base indicate the elements and their sequence in an entry in the main part
Example :	For an abstracting periodical:Each entry in the main part gives name(s) of author(s), title of paper, title of source periodical, volume and issue numbers, year, month, inclusive pages, language of the document, abstract and name of abstractor or source of the abstract
Indexed ? :	N
Remarks:	Optional

Field Tag:	820
Field Name:	Online access
Applicable to:	Document. System
Worksheet:	DIVPP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	Enter names in angular brackets ... of data bases/online services in which the document or information system/source is included
Example :	BRS DIALOG
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Whole
Remarks:	Optional

Field Tag:	821
Field Name:	Document delivery information
Applicable to:	Document. System
Worksheet:	DIVPP. ABNSY
Repeatable ? :	Y
Subfields:	None
Representation:	Especially for an indexing/abstracting serial, data base or information service indicate whether hard copies of the documents indexed/input are made available and related conditions, charges, etc.
Example :	Tear sheets and photocopies. US$1.00 per page including postage
Indexed ? :	N
Remarks:	Optional.

Field Tag:	902
Field Name:	MFNs of Publications
Applicable to:	Insitution. Project. System. Activity
Worksheet:	ABNIN. ABNRP. ABNSY. DIVAC
Repeatable ? :	Y
Subfields:	None
Representation:	Use the MFN of the bibliographic record which must be in the same database as the record being entered
Example :	A research project has published 2 documents which have been entered in the database at MFN 789 and MFN 790. Enter in field 902: 789%790
Indexed ? :	N

Field Tag:	950
Field Name:	Current status of project.
Applicable to:	Project.
Worksheet:	ABNRP
Repeatable ? :	Y
Subfields:	None.
Representation:	In free form or in coded form. A database may set up a set of codes e.g. A Completed; B In abeyance; C Continuing; D Abandoned; R Under revision, revised. The date may be added.
Example :	A project is abandoned in January 1992. Enter: D, Jan 1992

Field Tag:	954
Field Name:	Project number.
Applicable to:	Project.
Worksheet:	ABNRP
Repeatable ? :	Y
Subfields:	None.
Example :	SJ/88/123
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	955
Field Name:	Contract number.
Applicable to:	Project.
Worksheet:	ABNRP
Repeatable ? :	Y
Subfields:	None.
Representation:	According to the practices of the organization assigning the number
Example :	FED/1/456
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	957
Field Name:	Resources (equipment, etc.)
Applicable to:	Institution. Project.
Worksheet:	ABNIN. ABNRP
Repeatable ? :	Y
Subfields:	None.
Representation:	In free form. Enter details of resources excluding personnel enclosing terms to be indexed in angular brackets ...
Example :	HP3000 minicomputer 2%; IBM/XT microcomputer 6%; IBM/AT microcomputer 4
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Terms enclosed in angle brackets.

Field Tag:	960
Field Name:	Type of institution
Applicable to:	Institution
Worksheet:	ABNIN
Repeatable ? :	Y
Subfields:	None.
Representation:	In free form or codes agreed upon.
Example :	The database categorises institutions as G: Government; U: University; P: Private sector. Enter for a national research council G.
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole.

Field Tag:	961
Field Name:	Type of research
Applicable to:	Project
Worksheet:	ABNRP
Repeatable ? :	Y
Representation:	Research work may be categorized as Fundamental research, Applied research, Developmental research, Survey, etc., establishing/adopting an appropriate categorization
Example :	Applied research%Survey
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Remarks:	Useful in assessing the extent of and investment in different types research in an institution or in a country during a particular period.

Field Tag:	998
Field Name:	Location/Holdings
Applicable to:	Document
Repeatable ? :	Y
Subfields:	A, Institution code. B, Volumes held. C, Years covered. D, Notes
Representation:	For use especially in union catalogues and inventories/directories. Institution code: Establish/adopt an approrpiate code for the r of institutions to be included in the catalogue/directory; enter appropriate code in angular brackets .... Volumes held: Volumes (or numbers) of the periodical available in the institution. Years covered: the inclusive years corresponding to the volumes held. Notes: Indication of incompleteness of a volume, etc.
Example :	ROSTSEA, V. 1-5, 7-, 1965-1969, 1971-, V. 6 incomp.%; ROSTCA, V. 2-, 1966-
Indexed ? :	Y
Index Technique:	2
Whole or SB:	Subfield A.

Field Tag:	999
Field Name:	Record type
Applicable to:	All types of Records
Worksheet:	All worksheets
Repeatable ? :	N
Subfields:	None
Representation:	See MIBIS. Enter: B for all bibliographical records C for Corporate name authority record R for Serials authority recor X Supplier authority record E for record on expert/person I for record on institution P for record on project S for record on information system
Indexed ? :	Y
Index Technique:	0
Whole or SB:	Whole
Remarks:	Useful in retrieving any one type of record, e.g. all bibliographical records only; or institution profiles only, etc. Can be combined (Boolean) with other search terms.

ANNEX 4-2: Field Definition Table (ABNCD.FDT)

Field Definition Table (FDT) Data Base: ABNCD

?|	Tag|	Name	|Len	|Typ|	Rep|	Delimiters/Pattern
	1	Participating centre code	100	X		
	2	Participating centre record no	6	N		
	3	Record status	1	P		A
	5	Date record entered	10	P		9999-99-99
	6	Date record changed	10	P		9999-99-99
	7	Bibliographic level	5	A		
	8	Bibliographic level - parent	1	A		
	9	Country of origin of record	2	P		AA
	10	Record number of parent	6	N		
	11	Record number(s) of part(s)	6	N	R	
	12	Record no of other lang ver(s)	6	N	R	
	20	Language of analysis	18	A		
	21	Language(s) of text	2	A	R	
	22	Language(s) of summaries	2	A	R	
	25	Record heading	50	X		
	100	Title	500	X		
	101	Parallel title(s)	500	X	R	
	102	Translated title - English	500	X		
	105	Translated title - other	500	X		
	110	Personal author(s)	80	X	R	ab
	111	Corporate author(s)	500	X	R	abcdz
	112	Affiliation	500	X		abcdz
	113	Other associated inst(s)	500	X	R	abcdez
	114	Meeting	500	X	R	abcde
	115	Trans. name of instn.	200	X		
	116	Address	300	X	R	abcdefghi
	120	Edition	25	X		
	121	Publisher	250	X		abc
	122	Date of publ/issue - free form	30	X		
	123	Date of publ/issue - ISO form	10	P		9999-99-99
	130	Collation (M/C)	200	X		abc
	131	Part statement	150	X		ab
	140	Monographic series	200	X	R	abz
	141	Thesis	200	X		abcd
	142	Related project(s)	200	X	R	ab
	150	Notes	700	X		
	160	ISBN	13	X	R	
	161	Document number	50	X	R	
	162	Availability	100	X		
	200	Title of serial	400	X		z
	201	ISSN	9	P		9999-999X
	202	Title of parent (M/C)	500	X		
	210	Personal author(s) - parent	80	X	R	ab
	211	Corporate author(s) - parent	500	X	R	abcdz
	300	Primary descriptors	200	X		
	301	Secondary descriptors	400	X		
	302	Geographic descriptors	200	X		
	303	Local descriptors	200	X		

921	Supplier authority code	4	X		
922	Supplier name and address	200	X		abcde
997	Authority record notes	200	X		
998	Authority record date	10	P		9999-99-99
441	Duration	50	X		
442	Date:proposal/approval	25	X		ab
443	Date:starting	10	X		
444	Date:expect. compl.	10	X	R	
445	Date:actual compl.	10	X		
446	Date:terminated	10	X		
447	Date of birth	100	X		
830	Nationality	100	X	R	
831	Qualifications	100	X	R	abcd
832	Specialization	100	X	R	
833	Work experience (last)	200	X		abcde
834	Current work	200	X		abcde
835	Marital status	10	X	R	
836	Sex	6	X		
850	Recommended by	100	X	R	abcd
855	Honours and awards	200	X	R	abc
856	Membership in societies	200	X	R	abcd
525	Language competence	100	X	R	abc
556	Assignments	200	X	R	abcd
895	Databases	300	X	R	ndrfa
896	Classification system used	100	X	R	
897	Subject headings list	100	X	R	
898	Thesaurus	100	X	R	
899	Periodical publicat.	300	X	R	ij
890	Patents taken	200	X	R	abcdefgh
900	Services offered	200	X	R	
570	Personnel	100	X	R	ab
625	Objectives	500	X	R	
700	Financial aspects	200	X	R	sacp
950	Project status	50	X		
952	Training courses	200	X	R	
954	Project number	50	X	R	a
955	Contract number	50	X	R	
957	Resources(equipment...)	200	X	R	
960	Type of institution	100	X	R	
961	Type of research	100	X	R	
965	Research priority	100	X		
966	Commitee's decision	100	X		
999	Record type	1	P		A
1000	Name of object	100	X		
1001	Local name (Eng.)	100	X	R	
1010	Function	300	X	R	
1015	Source/Donor (Person)	100	X	R	sfh
1016	Source/Donor (Organization)	300	X	R	

514	Requester	25	X	R	
515	Supplier	200	X		abcdez
516	Price	20	X		ab
517	Acquisition notes	200	X	R	
901	Corporate body	500	X		abcd
902	See reference(s)	500	X	R	
903	Other language version(s)	500	X	R	
904	Former name(s)	500	X	R	
905	Later name(s)	500	X	R	
908	Reference code	20	X		
911	Serial title	400	X		
912	ISSN	9	P		9999-999X
913	See reference(s)	400	X	R	z
914	See also other lang edition(s)	400	X	R	
915	Former name(s)	400	X	R	
916	Later name(s)	400	X	R	
304	Proposed descriptors	100	X		
310	Abstract/Description	1000	X	R	
320	Broad subject heading	100	X		
400	Processing status	4	X		
410	Location	10	X	R	
411	Call number	40	X		
412	Number of copies	2	N		
415	Accession numb.	10	X		
420	Type of material	50	X		
430	Documentalist (initials)	10	X	R	
500	Acquisition type	4	X		
509	Order number	25	X		
510	Date ordered	10	P		9999-99-99
511	Date claimed	10	P		9999-99-99
512	Date received	10	P		9999-99-99
513	Number of copies ordered	2	N		
1017	Vendor	300	X		
1018	Price	100	X		cp
1020	Provenance	100	X		
1021	Archaeological site	500	X		
1025	Ethnic group	100	X	R	
1028	Date	100	X		
1030	Material	300	X	R	
1035	Condition	1000	X	R	
1040	Dimension (Front)	100	X		hwld
1041	Dimension (Back)	100	X		hwld
1042	Weight	100	X		
1050	Description	1000	X		
1055	Fine number	100	X		
1056	Photo number	100	X	R	
1060	Negative number	100	X	R	
1065	Accession number	100	X		

A - Insert (after) | B - Insert (before) | C - Change line | D - Delete line
P - Previous page | N - Next page | T - Top | E - Bottom
X - Exit | ↵ - Next line

ANNEX 4-3: DATA ENTRY ASSISTANCE (DEA.PAS) SOURCE CODE.

```
program DEA(s1:string) [menu];

var
    dbname,names,
    flds,fld,sold,
    term,s,t,f,a,keyw,ii,db : string;
    voc,soln,mtag,modb      : string;
    ntag,nfld,mfn,new,
    cur,dis1,fdt1,
    nfldold,code,
    msglng,tag,
    i,j,k,l,m,n,nn,rc,rno,add,th,nr: real;
    no,num,row,lin                 : real;
    num1,num2                      : real;
    ftag       : array [1..100] of real;
    flng       : array [1..100] of real;
    fbgn       : array [1..100] of real;
    tags       : array [1..100] of real;

Function fuc(sm: string): string;
var us: string;
begin
us:=sm; uc(us);
fuc:=us;
end;


Procedure THESHI;

var dt: array[1..15] of real;
    doc: array[1..15] of real;
    rn: array[1..15] of real;
    tag: array[1..10] of real;    { tag of relation }
    maxt: real;                  { max no. of tags (upper bound of
                                                          tag) }
    maxl: real;                 { max no. of lines (upper bound of
                                                    dt,doc) }
    rel,invrel: string;           { Relation indicators }
    nl: real;                     { lines on this page }
    cl: real;                     { current line }
    term: string;                 { current term }
    q: string;                    { query }
    dbname: string;               { current data base }
    mfn: real;                    { current mfn }
    tmfn: real;                   { mfn of current term }
    s,action,ft,answ: string;
    i,k,kl,lq,rc: real;
```

```
Procedure ERRMSG(t: string);

{ Display error message t and pause  }
var s: string;
begin
clearmsg; writeln(chr(7),t);
write('Press ENTER to continue'); s:=inkey;
end;

Procedure DISPLT;
begin
cleardata;
box(1,1,3,45,1); clearbox(2,2,1,43,2);
cursor(2,2); write(term);
if action='S' then
   begin
   box(1,74,3,7,1);
   cursor(1,76); write('MFN'); cursor(2,75); write(mfn:5);
   end;
savescr(1);
end;

Procedure GETREC(n: real);
var rc: real;
begin
if n<>mfn then
   begin
   mfn:=n; rc:=record(mfn);
   end;
end;

Procedure DISPLAY(t,o: real);
var rc,fn: real;
    it,io: real;                    { current tag/occ }

Procedure Disprel(it,io,fn,mfr,level: real);
var rc,k,relmfn,rfn,i: real;
begin
nl:=nl+1; dt[nl]:=it; doc[nl]:=io; rn[nl]:=mfr;
getrec(mfr);
cursor(nl+4,1); write('_ ',substr(rel,(it-1)*3+1,3));
if level>1 then write(level:1);
cursor(nl+4,level+7); write(field(fn));
if (it=5) or (it=6) then
if find(fuc(field(fn)))=0 then
   begin
   k:=nxtpost; relmfn:=posting('MFN');
   getrec(relmfn);
   rfn:=fieldn(tag[it],1); i:=1;
   while (rfn>0) and (nl<maxl) do
     begin
```

```
      disprel(dt[nl],i,rfn,relmfn,level+1);
      getrec(relmfn);
      i:=i+1; rfn:=fieldn(tag[it],i);
      end;
    end;
end;
begin
nl:=0;
if t=1
   then begin
        displt;
        it:=1; io:=1;
        end
   else begin
        clearbox(5,1,15,80,0);
        it:=t; io:=o;
        end;
while (it<=maxt) and (nl<maxl) do
  begin
  repeat
    fn:=fieldn(tag[it],io);
    if fn=0 then begin it:=it+1; io:=1; end;
  until (fn>0) or (it>maxt);
  if fn>0 then disprel(it,io,fn,mfn,1);
  getrec(tmfn);
  io:=io+1;
  end;
end;

Function DECIDE(l: real): string;
var s: string;
begin
cl:=l;
if nl>0 then
begin
clearmsg;
writeln('<SpB>[Next] B[ack] F[irst] P[age] S[elect] T[erm select]
Q[uery] M[ark term(s)]');
write('?[Disp query] A[dd rel] D[elete] C[reate term]
W[rite in Fld] X[exit]');
  repeat
  if cl<1 then cl:=1;
  if cl>nl then cl:=nl;
  cursor(cl+4,1);
  s:=fuc(inkey);
  case s of
  ' ': if cl>=nl then cl:=1 else cl:=cl+1;
  'B': begin cursor(cl+4,1); write(''); cl:=cl-1; end;
  'F': begin display(1,1); cl:=1; end;
  'P': begin
       getrec(rn[nl]);
       display(dt[nl],doc[nl]);
```

```
        cl:=1;
        end;
   end;
   until position('?ACDLSQMTWGX',s,1)>0;
end;
decide:=s;
if s='S' then
   begin
   getrec(rn[cl]);
   term:=field(fieldn(tag[dt[cl]],doc[cl]));
   end;
end;

Function FINDTERM(term: string): real;
var rc: real;
    t: string;
begin
t:=fuc(term);
rc:=find(t);
findterm:=rc;
if rc=0 then
   if nxtpost<0
      then findterm:=1
      else begin
           mfn:=posting('MFN');
           rc:=record(mfn);
           findterm:=rc;
           if rc=0 then
              begin
              tmfn:=mfn;
              display(1,1);
              action:=decide(0);
              end;
           end;
end;

Function FLDUC(k: real): string;

{ Returns k-th field of record converted to upper case }

var f: string;
begin
f:=field(k); uc(f);
flduc:=f;
end;

Function CHKREL(t: string): real;

{ Check if a relation already exists }

var i,n: real;
begin
```

```
n:=nfields; i:=1;
while (i<=n) and (flduc(i)<>t) do i:=i+1;
if i>n then chkrel:=0
       else chkrel:=i;
end;


Procedure UPDINVF;

{ Update inverted file (screen is clear because FST is displayed)}

begin
cleardata;
updif;
end;


Procedure CREATERM;

{ Create new thesaurus term }

var tuc: string;
    rc,np: real;
begin
term:=''; clearmsg;
displt;
clearmsg; write('Enter new term');
rc:=edit(term,30,2,2,30,1,' ');
if term<>'' then
   begin
   tuc:=term; uc(tuc); rc:=find(tuc); np:=-1;
   if rc=0 then np:=nxtpost;
   if (rc=0) and (np>0)
      then errmsg('Term already exists')
      else begin
           mfn:=newrec;
           rc:=fldadd(tag[1],1,term);
           update; updinvf;
           action:='S';
           end;
   end
   else action:='T';
end;

Procedure ADDREL;

{ Add new relation to a term  }

var r,rt,rtu: string;
    rc,i,rtag: real;

Function ADDIT: real;
```

```
var tt,ir: string;
    n,k: real;
    relmfn: real;

Procedure RELADD;
var rc: real;
begin
n:=nocc(rtag); k:=1;
while (k<=n) and (flduc(fieldn(rtag,k))<rtu) do k:=k+1;
rc:=fldadd(rtag,k+1,rt); update;
end;

begin
if (find(rtu)<>0) and (substr(r,1,2)<>'SN')
   then begin
        addit:=1;
        errmsg('Related term does not exist');
        end
   else
if (chkrel(rtu)<>0) and (substr(r,1,2)<>'SN')
   then begin
        addit:=1;
        errmsg('Relation already exists');
        end
   else
begin
rtag:=tag[(rtag-1)/3+1];
reladd;
ir:=substr(invrel,(rtag-1)*3+1,3);
if ir<>'   ' then
   begin
   k:=nxtpost; relmfn:=posting('MFN');
   rtag:=tag[(position(rel,ir,1)-1)/3+1];
   rt:=field(fieldn(tag[1],1)); rtu:=rt; uc(rtu);
   k:=record(relmfn);
   reladd;
   end;
k:=record(mfn);
addit:=0;
end;
end;

begin
box(18,10,3,5,1); box(18,14,3,52,1);
cursor(19,1); write('Relation');
r:=''; rt:='';
repeat
clearbox(19,15,1,50,1);
clearmsg; write('Enter relation code: ');
for i:=2 to maxt do write(substr(rel,(i-1)*3+1,3),' ');
clearbox(19,11,1,3,1); rc:=edit(r,3,19,11,3,1,' '); uc(r);
rtag:=position(rel,r,1);
```

```
if rtag=0 then write(chr(7));
until (r='') or (rtag>0);
repeat
i:=0;
if rtag>0 then
   begin
   clearmsg;
   rc:=edit(rt,50,19,16,50,1,' ');
   rtu:=rt; uc(rtu);
   if rtu<>'' then i:=addit;
   end;
until i=0;
action:='S';
end;

Procedure DELREL;
{ Delete a relation  }

var rtag,rc,k,relmfn: real;
    rt,rtu,ir: string;
begin
rtag:=fieldn(dt[cl],doc[cl]);
rt:=field(rtag); rtu:=rt; uc(rtu);
rc:=flddel(rtag);
update;
ir:=substr(invrel,(dt[cl]-1)*3+1,3);
if ir<>'   ' then
   begin
   rc:=find(rtu);
   if rc=0 then
      begin
      k:=nxtpost;
      if k>=0 then
         begin
         relmfn:=posting('MFN');
         rtag:=tag[(position(rel,ir,1)-1)/3+1];
         rt:=field(fieldn(tag[1],1));
         rtu:=rt; uc(rtu);
         rc:=record(relmfn);
         if rc=0 then
            begin
            k:=chkrel(rtu);
            if k>0 then
               begin
               rc:=flddel(k);
               update;
               end;
            end;
         end;
      end;
   end;
k:=record(mfn);
```

```
action:='S';
end;


Procedure DELTRM;


{ Delete a thesaurus term }

begin
if nfields>1
   then begin
        errmsg('Cannot delete term with relations. Delete all
relations first.');
        action:='S';
        end
   else begin
        rc:=flddel(1);
        update; updinvf;
        action:='T';
        end;
end;

begin
maxt:=7;
rel:=    '    SN USEUF PB SB RB ';
invrel:='       UF USESB PB RB ';
for i:=1 to maxt do tag[i]:=i;
maxl:=15; q:='';
dbname:=MODB;
If dbname <> '' then
begin
open(dbname);
clear;
if maxmfn=1 then action:='C' else action:='T';
repeat
case action of
'T': begin
     clearmsg;
     write('Select term');
     term:=''; displt;
     cursor(2,2); readln(term);
     if (substr(term,size(term),1)='$') or (findterm(term)<>0) then
     action:='L';
     end;

'L': begin
     page(1);
     uc(term);
     if (term='') or (substr(term,1,1)<'!') then term:='!';
     rc:=find(term);
     repeat
```

```
          ft:=term;
          i:=5;
          repeat
            cursor(i,5); write('-  ',term);
            term:=nxtterm; i:=i+1;
          until (i=21) or (term='');
          k:=5; rc:=find(ft);
          repeat
            cursor(k,5); action:=fuc(inkey); k:=k+1;
            if action<>'S' then ft:=nxtterm;
          until (position('PSTX',action,1)>0) or (k=i);
          if (action='P') and (k<i)
             then repeat ft:=nxtterm; k:=k+1; until k=i;
          term:=ft; page(1);
        until (position('STX',action,1)>0) or (ft='');
        if term='' then action:='L';
        end;
'S': begin
        rc:=findterm(term);
        if rc<>0 then action:='L';
        end;
'A': { Add a relation }

        addrel;

'C': { Create a new term }

        createrm;

'D': { Delete a term or a relation }

        if cl=1 then deltrm else delrel;

'Q': { Select term for searching }

        begin
        s:=field(fieldn(tag[dt[cl]],doc[cl]));
        if size(s)+size(q)+3>255
           then begin
                write('');
                action:='?';
                end
           else begin
                if q<>'' then q:=q|' +  ';
                q:=q|s;
                action:=decide(cl+1);
                end;
        end;


'M': { Mark term(s) for inclusion in  field }
```

```
          ft:=term;
          i:=5;
          repeat
            cursor(i,5); write('-  ',term);
            term:=nxtterm; i:=i+1;
          until (i=21) or (term='');
          k:=5; rc:=find(ft);
          repeat
            cursor(k,5); action:=fuc(inkey); k:=k+1;
            if action<>'S' then ft:=nxtterm;
          until (position('PSTX',action,1)>0) or (k=i);
          if (action='P') and (k<i)
             then repeat ft:=nxtterm; k:=k+1; until k=i;
          term:=ft; page(1);
        until (position('STX',action,1)>0) or (ft='');
        if term='' then action:='L';
        end;
  'S': begin
        rc:=findterm(term);
        if rc<>0 then action:='L';
        end;
  'A': { Add a relation }

        addrel;

  'C': { Create a new term }

        createterm;

  'D': { Delete a term or a relation }

        if cl=1 then deltrm else delrel;

  'Q': { Select term for searching }

        begin
        s:=field(fieldn(tag[dt[cl]],doc[cl]));
        if size(s)+size(q)+3>255
           then begin
                write('');
                action:='?';
                end
           else begin
                if q<>'' then q:=q|' +  ';
                q:=q|s;
                action:=decide(cl+1);
                end;
      end;


  'M': { Mark term(s) for inclusion in  field }
```

```
        begin
        s:=field(fieldn(tag[dt[cl]],doc[cl]));
        if tag[dt[cl]]=1 then s:='^a'|s;
        if tag[dt[cl]]=2 then s:='^b'|s;
        if tag[dt[cl]]=6 then s:='^r'|s;
        if size(s)+size(q)+3>255
           then begin
                write('');
                action:='W';
                end
           else begin
                if q<>'' then q:=q|'';
                q:=q|s;
                action:=decide(cl+1);
                end;
      end;

  'W':{ Add term(s)/code to a particular field}
      begin
        KeyW:=q; clear;
        If KeyW <>'' then
         begin
           box(09,1,10,80,2);cursor(10,5);
           writeln('Term(s) selected: ',KeyW);
         end
         else writeln('Term(s) selected: NONE ');
        cursor(12,5);
        cursor(13,5);write('Do you want to continue -- Y or N : ');
        read(answ); uc(answ);
        If answ ='Y' then
         begin
           cursor(15,2);
           write('Which Master file number (MFN)? ');
           read(num);
           cursor(16,2);
           write('Which field?  ');
           read(num1);
           action:='X'
         end
        else
          begin
           action :='T';
           q :='';
          end;
    end;

  '?': { Display current query }

      begin
        savescr(2);
      box(16,8,6,66,2); clearbox(17,9,4,64,1);
      cursor(17,9); lq:=size(q);
```

```
      if lq=0 then write('No search terms currently selected') else
         begin
         k:=1; kl:=17;
         repeat
         if lq>64 then i:=64 else i:=lq;
         writeln(substr(q,k,i));
         k:=k+i; lq:=lq-i;
         kl:=kl+1; cursor(kl,9);
         until lq=0;
         end;
      clearmsg; write('Press any key to continue');
      s:=inkey;
       page(2);
       action:=decide(cl+1);
      end;
end;
until action='X';
close;
end
else
begin
 cursor(14,30);write('No Authority List Selected');
end;
end;

{=== Theshi procedure ends here ===}

procedure TEST;
   var aa:string;
       ii:real;
   begin
      savescr(4); clear;
      writeln('nfld=',nfld:1,', dis1=',dis1:1,', cur=',cur:1);
      writeln(' tag=',tag:1,', fld=',fld:50);
      write  ('----+----1----+----2----+----3----+----4',
              '----+----5----+----6----+----7----+----8');
      writeln(flds);
      write  ('----+----1----+----2----+----3----+----4',
              '----+----5----+----6----+----7----+----8');
      for ii:=1 to nfld+1 do
      begin
         write(ftag[ii]:1,'.',fbgn[ii]:1,'.',flng[ii]:1,' / ');
      end;
      writeln; write('^Q┘'); aa:=inkey; page(4);
   end;


procedure HILIT;
   begin
      chattr(1,5+cur,1,80);
   end;
```

```
procedure LOLIT;
   begin
      chattr(0,5+cur,1,80);
   end;
procedure HELPKEY;
   begin
      savescr(2);
      box(2,45,16,36,2); clearbox(3,46,14,34,1);
      cursor( 3,46); write('Key        Function');
      cursor( 4,46); write('--- ----------------------------');
      cursor( 5,46); write(' A  insert field After current');
      cursor( 6,46); write(' B  insert field Before current');
      cursor( 7,46); write(' C  Cancel changes & exit');
      cursor( 8,46); write(' D  Delete current field');
      cursor( 9,46); write(' E  Edit current field');
      cursor(10,46); write(' F  change current Field tag');
      cursor(11,46); write(' N  Enter new record');
      cursor(12,46); write(' X  store record & eXit');
      cursor(13,46); write('^X, PgUp, ^Y, PgDn ');
      cursor(14,46); write('Home, End   Scroll');
      cursor(15,46); write('Ctrl-Home,');
      cursor(16,46); write('Ctrl-End    Set screen top/bottom');
      a:='Press any key to return '; m:=size(a)+2;
      cursor(22,02); write(a:70); cursor(22,m); a:=inkey;
      page(2);
   end;

procedure SHOWFDT;
   var ii,jj : real;
   begin
      ii:=fdt1; jj:=5; clearbox(jj,46,12,34,1);
      repeat
         cursor(jj,46);
         write(tags[ii]:3,' ',substr(names,ii*30-29,30));
         ii:=ii+1; jj:=jj+1;
      until (ii>ntag) or (jj>16);
      while jj<17 do begin cursor(jj,46); write(' ':34); jj:=jj+1;
end;
      a:='Scroll with PgDn,PgUp,^Y,^X and Home; Other keys return
'; m:=size(a)+2;
      cursor(22,2); write(a:70);
   end;


procedure HELPFDT;
   begin
      savescr(1); box(2,45,16,36,1); clearbox(3,46,2,34,1);
      cursor(3,46); write('Tag       Name of field          ');
      cursor(4,46); write('--- ----------------------------');
      showfdt;
      repeat
         cursor(22,m); code:=kbdkey(a);
```

```
            if ord(a)>0 then begin write(a); uc(a); code:=256+ord(a);
    end;
            if (code=72) or (code=75) then code:=73 else
            if (code=77) or (code=80) then code:=81;
            if (code<>71) and (code<>73) and
               (code<>80) and (code<>81) then code:=0;
            case code of
                0: page(1);
               71: if fdt1 > 1 then
                      begin fdt1:=1; showfdt;
                      end; {home}
               73: if fdt1 > 1 then
                      begin fdt1:=fdt1-10; showfdt;
                      end; {up}
               80: begin
                      k:=(ntag-1)/12; a:=encint(k,1); k:=val(a)*12+1;
                      if fdt1<>k  then
                         begin fdt1:=k; showfdt;
                         end; {end}
                   end;
               81: if fdt1<ntag-9 then
                      begin fdt1:=fdt1+10; showfdt;
                      end; {down}
            end;
         until code=0;
      end;
   procedure SHOWFLD(cur2:real);

   var kk,ll : real;
   begin
      kk:=dis1+cur2; ll:=flng[kk]; if ll>71 then ll:=71;
      s:=encint(ftag[kk],1); while size(s)<3 do s:='0'|s;
      s:=encint(kk,1)|' '|s; while size(s)<6 do s:='0'|s;
      cursor(5+cur2,2); write(s|' '|substr(flds,fbgn[kk],ll));
   clearln;
   end;

   procedure SHOWREC(cur2:real);
   var ii,jj,kk,ll : real;
   begin
      if cur2<0 then jj:=-1 else
      begin
         if dis1+9<=nfld then jj:=9 else jj:=nfld-dis1;
         for ii:=cur2 to jj do
         begin
            kk:=dis1+ii; ll:=flng[kk]; if ll>71 then ll:=71;
            {if mtag='0' then
            begin}
              s:=encint(ftag[kk],1);
              while size(s)<3 do s:='0'|s;
            {end
            else s:=mtag;}
```

```
            s:=encint(kk,1)|' '|s; while size(s)<6 do s:='0'|s;
            cursor(5+ii,2); write(s|' '|substr(flds,fbgn[kk],11));
            clearln;
         end;
      end;
      if jj<9 then clearbox(6+jj,1,9-jj,80,0);
      s:='F1 Lists FDT Fields, F2 Lists Functions.  NEXT FUNCTION: ';
      msglng:=size(s)+2;
      cursor(22,2); write(s:70);
   end;

   function EDITAG(ifld:real) : real;
      var ii,jj,kk:real;
      begin
         if ifld=0 then s:='' else s:=encint(ftag[ifld],1);
         sold:=s;
         repeat
            kk:=edit(s,5,16,9,5,1,'_');
            if kk=1 then helpfdt else
            if kk=4 then begin s:=sold; kk:=1; end else
            begin
               ii:=val(s); jj:=0; kk:=1;
               if ii=0 then jj:=ntag;
               while (jj<ntag) and (kk=1) do
               begin jj:=jj+1; if ii=tags[jj] then kk:=0; end;
               if kk=1 then begin cursor(24,2); write('Invalid
tag.^G':70); end;
            end;
         until kk<>1;
         Editag:=ii;
        attr('_',1,16,9,5);
      end;

   procedure ShowMsg;

   var
      int,m,rw,fm,next,sc : real;
      str1,ans,key        : string;

   begin
      open(MODB);
      savescr(3);
      int:=find(term);
      if int = 0 then
       begin
       autotype(' ');
       sc:=search(term);
       ans:=inkey;
       m:=setpos(0,1);
       end;
       clear;
       page(3);
```

```
      int:=record(m);
      clearbox(5,35,10,45,1);
      box(5,35,10,45,1);
      rw:=6;
      if int = 0 then
      fm:=format(44);
      if fm = 0 then
    begin
      next:=nxtline(str1);
      while (next = 0) and (soln <>'N') do
      begin
        cursor(rw,36);
        writeln(str1);
        next:=nxtline(str1);
        rw:=rw+1;
        if (rw = 13) and (next = 0) then
        begin
          cursor(14,52); write('.. More..  Y[es]  N[o]');
          repeat
            soln:=fuc(inkey);
          until (soln='Y') or (soln='N');
          if (soln='Y') then
          begin
            clearbox(6,36,7,43,1);
            rw:=6;
          end;
        end;
      end;
    end;
end;

Function EDIFLD(ifld:real) : string;
   var k,kk:real;
       ans:string;
begin
      attr(' ',0,16,9,5); s:=encint(tag,1);
      while size(s)<3 do s:='0'|s; write(s); term:=s;
        if (ifld=0) then s:='' else
        s:=substr(flds,fbgn[ifld],flng[ifld]);
        cursor(23,50); write('Consult guide? ... Y/N ');
        repeat
          ans:=fuc(inkey);
        until (ans = 'Y') or (ans = 'N');
        if ans = 'Y' then
        begin
          clearmsg;
          clearbox(5,38,10,43,1);
          box(5,38,10,42,2);
          cursor(6,40); write('Available Data Entry Guides');
          cursor(7,40);
          write('----------------------------');
          cursor(8,40);
```

```
            write('AGRIS - AGRIS subject codes');
            cursor(9,40);
            write('AGRO  - FAO AGROVOC    ');
            cursor(10,40);
            write('AUF   - Authority List(s)');
            cursor(11,40);
            write('GUIDE - Applicable to ABNCD, MIBIS');
            cursor(12,40);
            write('THESB - Information retrieval thesaurus');
            cursor(13,40);
            write('THES  - OECD Macrothesaurus ');
            cursor(23,50); write('Which guide?  ');
            readln(modb); uc(modb);
            if ((MODB<>'') and (MODB<>'AUF') and (MODB<>'THESB') and
            (MODB<>'THES') and (MODB<>'AGRIS') and (MODB<>'AGRO'))
             then
               begin
                 ShowMsg;
                 open(db);
                 rc:=record(rno);
                    begin
                      if size(s)>255 then
                         begin
                           cursor(24,2);
                           write('field length > 255.^G':70);
                         end
                      else
             begin
               cursor(23,20); clearLn;
               write('Edit it further, or press <ENTER> to accept');
               sold:=s;
                repeat
                  kk:=edit(s,255,18,9,232,1,'_');
                   if kk=1 then helpfdt else
                   if kk=4 then begin s:=sold; kk:=1; end;
                until kk<>1;
          end;
         EDIFLD := s;
         if nr = 1 then
           begin
             rno :=mfn;
             rc:=record(rno);
             kk :=fldadd(tag,ifld,s);
             update;
             rc:=record(rno);
           end;
       attr('_',1,18,9,232); attr('_',1,16,9,5);
     end; clearmsg;SHOWREC(0);
end
else
begin
  THESHI;
```

```
      open(db);
      rc:=record(num);
       if rc = 0 then
         begin
            num2 :=fldadd(num1,nfields+1,KeyW);
            if num2 = 0 then update;
          end;
end;
end
 else
    begin
        if size(s)>255 then
       begin
         cursor(24,2);
         write('field length > 255.^G':70);
       end
       else
         begin
           Cursor(23,20);
           write('Edit it further, or press <ENTER> to accept');
           sold:=s;
              repeat
                kk:=edit(s,255,18,9,232,1,'_');
                 if kk=1 then helpfdt else
                 if kk=4 then begin s:=sold; kk:=1; end;
              until kk<>1;
         end;
        EDIFLD := s;
        if nr = 1 then
          begin
            rno :=mfn;
            rc:=record(rno);
            kk :=fldadd(tag,ifld,s);
            update;
            rc:=record(rno);
          end;
      attr('_',1,18,9,232); attr('_',1,16,9,5);
     end; clearmsg;SHOWREC(0);
end;

procedure DELFLD(ifld:real);
   var ii,jj,ll : real;
   begin
      if ifld=nfld then flds:=substr(flds,1,fbgn[ifld]-1)
                   else flds:=substr(flds,1,fbgn[ifld]-1)|

      substr(flds,fbgn[ifld+1],fbgn[nfld+1]-fbgn[ifld+1]);
      ll:=flng[ifld];
      for ii:=ifld+1 to nfld do
      begin
         ftag[ii-1]:=ftag[ii];
         fbgn[ii-1]:=fbgn[ii]-ll;
```

```
            flng[ii-1]:=flng[ii];
         end;
         ftag[nfld]:=0;
         fbgn[nfld]:=fbgn[nfld+1]-ll; flng[nfld]:=0;
         nfld:=nfld-1;
      end;

   procedure addfld(tag,ifld:real; fld:string);
      var ii,jj,ll,k,kk : real;
      begin
         ll:=size(fld); nfld:=nfld+1;
         if ifld=nfld then  flds:=flds|fld
                      else flds:=substr(flds,1,fbgn[ifld]-1)|fld|

   substr(flds,fbgn[ifld],fbgn[nfld]-fbgn[ifld]);
         jj:=nfld-ifld;
         for kk:=1 to jj do
         begin
            ii:=nfld-kk;
            ftag[ii+1]:=ftag[ii];
            fbgn[ii+1]:=fbgn[ii]+ll;
            flng[ii+1]:=flng[ii];
         end;
         ftag[nfld+1]:=0; flng[nfld+1]:=0;
   fbgn[nfld+1]:=fbgn[nfld]+flng[nfld];
         ftag[ifld]:=tag; flng[ifld]:=ll;
      end;
   procedure repfld(ifld:real; fld:string);
      var ii,jj,kk,ll : real;
      begin
         if flds='' then flds:=fld else
         flds:=substr(flds,1,fbgn[ifld]-1)|fld|
               substr(flds,fbgn[ifld+1],fbgn[nfld+1]-fbgn[ifld+1]);
         ll:=size(fld); jj:=ll-flng[ifld];
         flng[ifld]:=ll;
         for kk:=ifld to nfld do
         begin
            ii:=kk+1; fbgn[ii]:=fbgn[ii]+jj;
         end;
      end;
   begin
      clear; cursor(1,29); write('DATA ENRTY ASSISTANCE');
      dbname:=dbn; fdt1:=1;
      ntag:=0; names:=''; mtag:='';
      if dbname='' then
      begin
         clearmsg; write(msgtext(1),' '); readln(dbname);
         open(dbname);
      end;
      db:=dbname;
      clearmsg; write(' ... Please wait ...');
      assign('Inp',path('dbn',10)|dbname|'.Fdt');
```

```
   repeat readln(inp,s) until substr(s,1,1)='*';
   while not eof(inp) do
   begin
      readln(inp,s); names:=names|substr(s,1,30);
      i:=position(s,' ',51); f:=substr(s,51,i-51);
      ntag:=ntag+1; tags[ntag]:=val(f);
   end;
   cursor(01,02); write('DATABASE:');
   attr(' ',1,1,12,size(dbname)); write(dbname);
   db:=dbname;
   cursor(01,69); write('MFN:'); attr(' ',1,1,74,6);
   cursor(03,02); write('No Tag Content of field');
   cursor(04,02);
   write('-- -------+----1----+----2----+----3----+----4');
   write('----+----5----+----6----+----7-');
   cursor(15,02);
   write('-- -------+----1----+----2----+----3----+----4');
   write('----+----5----+----6----+----7-');
   cursor(16,04); write('Tag:');
   cursor(18,02); write('Field:');
   attr('_',1,16,9,5); attr('_',1,18,9,232);
   clearmsg;
   repeat
      a:='X - exit     N - new record     MFN of existing record
--^P ';
      l:=size(a)+3;
      cursor(22,2); write(a:70); cursor(22,l-1); s:=inkey; uc(s);
      cursor(24,2); write(' ':70);
      flds:=''; nfld:=0;
      if s='X' then MFN:=-1 else
      if s='N' then begin MFN:=maxmfn; new:=1; nr:= 1; end else
      begin
         new:=0;
         cursor(22,1); readln(f); s:=s|f; mfn:=val(s); rno :=mfn;
         if mfn>=maxmfn then mfn:=0;
         if mfn=0 then
         begin
            cursor(24,2); write('Invalid mfn.^G':70);
         end
                  else
         begin
            k:=Record(mfn);
            if k<>0 then
            begin
               cursor(24,2); write('deleted record.^G');
               if k=1 then write(' Undelete? [Y/N]: ')
                      else write(' Reuse? [Y/N]: ');
               f:=Inkey; uc(f); if f='Y' then k:=0;
               cursor(24,2); write(' ':70);
            end;
            if k=0 then
            begin
```

```
71: begin
       lolit;
       if dis1<>1 then begin dis1:=1; showrec(0);
    end; { home }
       if cur  >0 then cur:=0;
       hilit;
    end;
72: begin {Up}
       if dis1+cur>1 then
        begin
          lolit;
          if cur>0 then cur:=cur-1
           else begin dis1:=dis1-1;
           showrec(0); end;
          hilit;
       end
        else write('^G');
    end;
73: begin {PgUp}
       if dis1>1 then
          begin
          i:=dis1-10;
          lolit;
          if i>0 then       dis1:=i
                 else begin dis1:=1; cur:=0; end;
          showrec(0); hilit;
       end
                 else write('^G');
    end;
79: begin  {End}
       i:=nfld-9; if i<1 then i:=1;

       lolit;
       if dis1<>i then begin dis1:=i; showrec(0);
     end;
       i:=nfld-dis1;
       if cur<>i then cur:=i;
       hilit;
    end;
80: begin {Down}
       if dis1+cur<nfld then
         begin
          lolit;
          if cur<9 then cur:=cur+1
                   else begin dis1:=dis1+1;
                  showrec(0); end;
          hilit;
       end
       else write('^G');
    end;
81: begin {PgDn}
       i:=dis1+10;
```

```
        if i<=nfld then
        begin
           lolit;
           dis1:=i; showrec(0);
           if dis1+cur>nfld then cur:=nfld-dis1;
           hilit;
        end
                  else write('^G');
     end;
117: begin
        if cur<9 then   {Ctrl End }
        begin
           lolit;
           dis1:=dis1+cur-9; if dis1<1 then dis1:=1;
           showrec(0);
           if dis1+9<=nfld then cur:=9 else
           cur:=nfld-dis1;
           hilit;
        end;
     end;
119: begin
      if cur>0 then   { Ctrl Home }
        begin
           lolit;
           dis1:=dis1+cur; cur:=0; showrec(0);
           hilit;
        end;
     end;

321: begin                                  {add after}
        i:=dis1+cur+1; add:=1;
        if new = 1 then
          begin
            mfn :=newrec;
            new :=2;
          end;
        tag:=editag(0); fld:=edifld(0);
        if fld<>'' then
        begin
           addfld(tag,i,fld);
           lolit;
           if cur=9 then
           begin
             dis1:=dis1+10; cur:=0;
           end
            else cur:=cur+1;
           showrec(cur);
           hilit;
        end;
      showrec(cur);
     end;
```

```
322: begin
        add:=1;
        if nfld=0 then begin dis1:=1; cur:=0; end;
        i:=dis1+cur;
        if new = 1 then
        begin
         mfn :=newrec;
         new :=2;
        end;   { add before }
        tag:=editag(0); fld:=edifld(0);
        if fld<>'' then
        begin
           addfld(tag,i,fld); lolit; showrec(cur);
           hilit;
        end;
     end;
324: begin           { delete }
       i:=dis1+cur;
       lolit; delfld(i);
        if i>nfld then
        begin
           if cur=0 then
           begin
              if nfld=0 then showrec(-1)
                        else
              begin
                 dis1:=dis1-10; if dis1<1 then
                 dis1:=1;
                 cur:=9; while dis1+cur>nfld do
                 cur:=cur-1;
                 showrec(0);
              end;
           end
                    else
           begin
              cur:=cur-1; showrec(cur);
           end;
        end
                  else showrec(cur);
        if nfld>0 then hilit;
     end;
325: begin                                  {edit}
        i:=dis1+cur;
        tag:=ftag[i];
        fld:=edifld(i);
        lolit;
        if fld='' then
        begin
           delfld(i);
           if i>nfld then
           begin
              if cur=0 then
```

```
                        begin
                          dis1:=dis1-10; cur:=9;
                          showrec(0);
                         end
                          else begin
                            cur:=cur-1;
                            showrec(cur);
                            end;
                     end
                     else showrec(cur);
                end
                 else
                begin
                   repfld(i,fld); showfld(cur);
                end;
                hilit;
           end;
      326: begin
                i:=dis1+cur;   { field tag }
                ftag[i]:=editag(i);
                lolit; showfld(cur); hilit;
           end;
      334: begin                              { new }
                i:=dis1+cur+1; add:=1;
                if new = 1 then
                  mfn :=newrec;
                  tag:=editag(0); fld:=edifld(0);
                  rc:=record(mfn);
                  if rc = 0 then
                     begin
                       j :=fldadd(tag,nfields+1,fld);
                       if j = 0 then update;
                     end;
                      rc:=record(mfn);
                      if rc = 0 then
                      showfld(cur);
                      end;
       344: begin
                j:=0;   { store,exit }
                for i:=1 to nfld do if ftag[i]=0 then j:=j+1;
                if j>0 then
                begin
                   savescr(3);
                   clearmsg;
                   write(' tags of ',j:1,
                   ' fields could not be matched in the
fdt.^G');
                   Cursor (23,2);
                   write('if you store the record, their
tags',
                   ' will be set to 000. Store? [Y/N]: ');
                   a:=Inkey; uc(a);
```

```
                        page(3);
                      end
                            else a:='Y';
                      if a='Y' then
                      begin
                       if nr <> 1 then
                       begin
                         if new=1 then mfn:=newrec
                                  else
                                 { if add <> 1 then}
                                  for i:=1 to nfldold do
                                  k:=flddel(1);
                           { if add <> 1 then}
                             for i:=1 to nfld do
k:=fldadd(ftag[i],i,substr(flds,fbgn[i],flng[i]));
                          update;
                         end;
                          {update;} code:=323;
                        end;
                    end;
            end;
          until code=323;
          clearbox(5,1,10,80,0); clearbox(1,74,1,6,1);
      end;
   until mfn<0;
  voc:=inkey;
   s1:=' ';
end.
```

CHAPTER 5

ONLINE HELP ON FORMAT ERRORS

5.1 INTRODUCTION

A good information storage and retrieval software should provide an adequate online guide to users to facilitate control of the output both on-screen (soft copy) and on printer (hard copy). For this purpose the system should have a powerful formatting language allowing users to define precisely the formatting requirements for database records. In addition, a context-sensitive online guide to formatting will help to supplement the formatting language of the system. Usually when a database is created in an information centre/data bank and similar institutions, display formats are also created along with the database. Most DBMS and text retrieval software packages including MicroIsis permit the specification of one or more display formats. One of these may be treated as a default format by the system.

Users may wish to define display format(s) to meet specific needs. Perera (1992) mentions the need for user to define his/her own display format in addition to what the system may provide. These situations include:

- viewing or printing a user defined set of fields different from the format provided by the system;

- sorting of the retrieved records according to users' needs; and
- using different display formats for the output of records to meet different requirements like catalogue cards, accession lists, etc.

MicroIsis provides extensive facilities to satisfy such requirements. Its formatting language allows the user to define display/print formats to display fields and other contents in any preferred format in terms of sequence of the fields and subfields, spacing between fields and subfields, punctuation, captions/headings etc. MicroIsis formatting language can be used to create display/print format(s) for either temporary or permanent use. Although a complex format can be built from simple formatting statements, the many facilities offered taken together can make the formatting language appear complex, unless one becomes quite familiar with their use Any deviation from the language prescribed can lead to an error. On using the format for display of records, the system checks the syntax and displays an error message in the form of an error code. It is not easy to memorize the error codes and their corresponding messages, so as to be able to correct the error. Also, unless one refers to the Mini-micro CDS-ISIS Reference Manual, the correct format statement may not be evident. Therefore, an online guide on the error codes, what each signifies, and how to correct them becomes essential.

An online help on format errors to guide end-users on how to handle different types of format errors occurring during the preparation of display/print formats has been attempted by developing a program (MOCDS.PAS) written in CDS-ISIS Pascal.

For each error code, the corresponding error message, what is implied, and how to correct the error with examples has been entered into a separate text file. These files are stored in the directory indicated by parameter 5 of SYSPAR.PAR. In the Information Retrieval Services menu (EXGEN), the system displays records according to the selected display format, on pressing option B (Browse) or option D (Display search results), as the case may be. If the format specification contains an invalid format statement (in relation to MicroIsis's formatting language) then an error message is displayed as shown in figure 5-1.

```
The controlled climate in the plant chamber and its
influence upon assimilation and transpiration.
Bosian, G.  1985.  p. 225-232, illus. (Methodology
of plant eco-physiology: proceedings of the
Montpellier Symposium) Incl. bibl.
KEYWORDS: Paper on: plant evapotranspiration.



*** Format error 99, PFT=CDS

 Edit format (Y/N)?
```

Figure 5-1: Error message screen

5.2 FORMAT ERRORS

As mentioned earlier, when interpreting and executing a format, MicroIsis performs a syntax analysis of the format to ensure that it conforms to the formatting language rules. Whenever MicroIsis detects an error in the format, it interrupts the formatting, and issues an error message in the form of an error code with the MOCDS program operating, the additional message 'Press F9 for help' also appears on the screen (see Figure 5-2).

```
Error  n [pe]

***  Format error n, PFT=name

 Edit format (Y/N)?          Press F9 for help
```

Figure 5-2: Modified error message screen

where:

n stands for error code number;

name is a name of the display format as defined by the user; and

pe is the possible error

Any output produced before the error is detected will appear before the message. Below (see Figure 5-3) is an example obtained by browsing record no. 2 of the CDS database.

```
The controlled climate in the plant chamber and its
influence upon assimilation and transpiration.
Bosian, G.  1985.  p. 225-232, illus. (Methodology
of plant eco-physiology: proceedings of the
Montpellier Symposium) Incl. bibl.
KEYWORDS: Paper on: plant evapotranspiration.



*** Format error 99, PFT=CDS

 Edit format (Y/N)?          Press F9 for help
```

Figure 5-3: Output from the format containing invalid command(s).

Pressing **Y** will return the user to the display format for editing. It is assumed, in this case, that the user is familiar with the format errors and correction procedures. Nevertheless, the message 'Press F9 for help' is also displayed in this screen and user can get online help.

Whenever the MOCDS program is called it automatically establishes a link with the corresponding error message file and user PFT file.

In general, each of the error message records consists of four main components. These are:

- Error code number: this provides an error code number for visual verification;
- Error message: provides brief explanation on the nature of the error invoked;
- Cause: gives the possible reason(s) as to why such an error has occurred; and
- Solution: this section provides guidance of what should be done to correct the invoked error. In some cases examples have been provided to further assist the user.

A list of the format error codes is given in annex 5-1 to this chapter.

5.3 MOCDS OPERATION

5.3.1 Installation of MOCDS

The MOCDS.PAS (source code), MOCDS.PCD (compiled version) and the eighteen text files of error messages are supplied in the same diskette. This compiled program file should be copied to the subdirectory where the CDS-ISIS Pascal programs used with the MicroIsis are stored, usually the subdirectory called PROG as indicated in parameter 1 of SYSPAR.PAR file. Thus, for example, if a library/information centre has installed the MicroIsis in a directory called ISIS with subdirectories SYS, MENU, MSG, DATA and PROG, the MOCDS.PAS and MOCDS.PCD files should be copied to the \ISIS\PROG subdirectory. This is done as follows:

Insert the diskette provided in drive A.

At the DOS prompt X:\> type cd ISIS and press the <ENTER> key. This will put the user in the directory where MicroIsis is installed. The following DOS prompt will be shown, X:\ISIS>. Now type cd PROG followed by pressing the <ENTER> key. The DOS prompt will now read as follows X:\ISIS\PROG>. At the prompt type COPY A:MOCDS.* and press the <ENTER> key. This will copy both the source code and the compiled version into the PROG subdirectory

of the ISIS directory. It should be noted here that MOCDS.PCD file is mandatory while MOCDS.PAS file is optional.

If some changes are desired to the MOCDS.PAS program, these can be done using any text editor (e.g. EDLIN, Turbo Pascal editor, etc.) and then recompiled using option A 'Advanced programming services (ISISPAS)' of the MicroIsis main menu followed by the C command. Then key in MOCDS at the prompt 'program name?' and press <ENTER> key to initiate compilation process. Then follow the steps mentioned above to copy this version to the system.

The error message files should be copied into the subdirectory in which the ISIS data bases are located, i.e. the subdirectory indicated by parameter 5 of SYSPAR.PAR (unless SYSPAR.PAR has been redirected using parameter 0, in which case user has to check this file to see where the data base files have been installed. A separate text file has been created for each of the eighteen error messages. These are: ER_1, ER_2, ER_8, ER_19, ER_20, ER_26, ER_28, ER_51, ER_53, ER_54, ER_55, ER_56, ER_58, ER_60, ER_61, ER_99, ER_101 and ER_102. If the data bases are stored in the subdirectory DATA, then copy the files for error messages into the subdirectory DATA as follows:

COPY A:\ER_* X:\ISIS\DATA

this will copy all the error message files into the

\ISIS\DATA subdirectory.

Modifying the Information Retrieval menu (EXGEN) (using option U - System Utility Services in the MicroIsis main menu), introducing a new line, say, H - Help on format errors and interfacing MOCDS program at option H in this submenu (EXGEN).

While displaying results by using option B 'Browse' or D 'Display search results' in this submenu (EXGEN), an error message will be issued if the display/print format contains invalid formatting statement(s). Pressing <F9> as indicated in the modified screen, will execute MOCDS program.

The modular approach has been used in developing MOCDS program. It is composed of a number of functions and procedures, each of which performs a distinct function(s) and a calling (main) program that coordinates them.

MOCDS main screen is divided into three parts as shown in figure 5-4. These are: the display/print format window, the editing window and the error code message window. MOCDS program allows toggling between the three different windows. The options available in each screen window and the functions they perform are described hereafter.

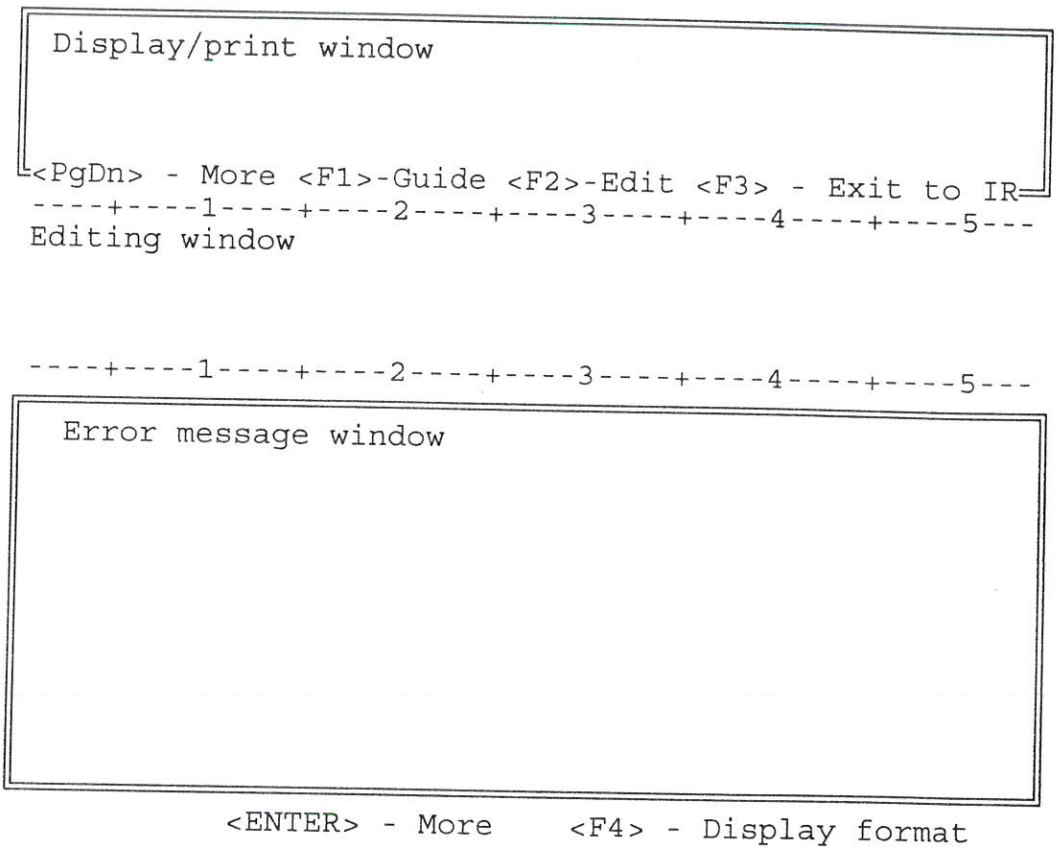


Figure 5-4: MOCDS main screen

5.3.2 Display/print format window

Display/print format window is the top most window of the MOCDS main screen and it is the part where the current format defined by the user during the current session will be displayed. This window is capable of displaying up to 240 characters, i.e. a maximum of six lines of 40 characters each. Since some of the display/print formats statements may extend over one page, i.e. over 1680 characters which is the MicroIsis page size, a srolling facility has been provided to allow for paging forward (Figure 5-4). MOCDS program allows

scrolling up to 30 screen pages, each screen is capable of holding upto 240 characters as mentioned above, a total of 7,200 characters can be accommodated by the program by assigning to the pre-defined array of string variables. Paging forward (Next page) is done by pressing a <PgDn> key. This default value of 7,200 characters has been reached under the assumption that, in general most of the user display/print formats may not exceed four pages and few may have access to extended memory (EMM) support. Nevertheless, MOCDS is flexible enough to accommodate EMM support. Furthermore, three other functions are available allowing user to consult online guide on format errors by pressing a function key <F1>, open an editing window by pressing a function key <F2> and exit back to the point where MOCDS program was called, i.e. at the information retrieval services menu by pressing a function key <F3>.

5.3.3 MOCDS editing window

MOCDS editing window is the middle window of the program's main screen. Having located the line in the display format by browsing through the format in the top window, pressing the function key <F2> will copy three lines including the error line into the middle editing window. In this window user may use all facilities provided by the MicroIsis field editor to perform

necessary correction(s). Immediately after correcting, the corrected version may automatically be saved to the user file with extension PFT by pressing <ENTER> key.

5.3.4 MOCDS error message window

This window is at the bottom of the MOCDS main screen. It is where the online guide on format errors displayed. Since the window is too small to display the whole text of a particular error code, scrolling facility has been provided to allow for paging forward by pressing the <ENTER> key.

Once the user is satisfied with the explanations offered, he/she can return back to the display/print window by pressing the function key <F4> indicated at the status bar.

5.4 USING MOCDS PROGRAM

MOCDS can be called only if one is in the information retrieval services menu, particularly when using display/print format containing invalid formatting statement(s). The system will issue an error code message as shown in figure 5-3.

User has three options at this point. These are:

- As indicated in figure 5-3, the user may terminate browsing session by simply typing **N** as a response

to the question: Edit format (Y/N)?. This option will retain the current version of the display/print format;

- In case the user intends to edit the display/print format he/she may press **Y** in the above question. Automatically the display/print format will be re-displayed as shown in figure 5-5. The cursor will be positioned at the top left-corner of the screen for editing. The following example is the CDS.PFT having an error 99.

```
Please enter/edit format (@xxxxx to use predefined
                                                format)
MDL, V12,V24, ,(|(|V76^Z|: |,V76^*|) |),V70+|; |,
V25,V26,V30,|(|V44|) |,V50,/|// |V71/|// |V72/|// |
V74/"KEYWORDS: ",V69(10,10)/##fi




EDIT: Replace                Press F9 for help
```

Figure 5-5: MicroIsis display/print format screen

- In case the user desires help at this stage , pressing F9 key will invoke MOCDS program.
- Alternatively if user is not familiar with the type of error reported by the system, for further information on how to deal with the error he/she may invoke help by pressing the function key **<F9>** as indicated in figure 5-3 (p. 168).

The error code will automatically be passed on to the MOCDS user interface and the corresponding error message will then be displayed along with the display/print format as shown in figure 5-6.

```
MDL,V12,V24,,(|(|V76^Z|: |,V76^*|) |),V70+|; |,V25,
V26,V30,|(|V44|) |,V50,/|//|V71/|// |V72/|// |V74/
"KEYWORDS: ",v69(10,10)/##fi

<PgDn> - More <F1>-Guide <F2>-Edit <F3> - Exit to IR
----+----1----+----2----+----3----+----4----+----5---
```

```
----+----1----+----2----+----3----+----4----+----5---
Error code: 99
--------------
Error message:
--------------
    Unknown command (e.g. misspelled function name
    or command); may also be due to a missing
    closing delimiter

          <ENTER> - More      <F4>-Display format
```

Figure 5-6: MOCDS help message window

At this point the cursor will be positioned at the first character of the display/print format window. A function key <PgDn> may be used as discussed in section 5.3.2. On pressing a function key <F1> the cursor will automatically be shifted to the bottom window. In this window user can scroll to get more information concerning the error. Once satisfied he/she may go back to the display/print format by pressing the function key <F4> as shown in the status bar in figure 5-6. While in the

display/print window, user can invoke the MicroIsis editor by pressing function key <F2>. As mentioned earlier, on pressing <F2>, MOCDS program will copy the text viewed in the top window to the middle window for editing as shown in figure 5-7.

```
MDL,V12,V24,,(|(|V76^Z|: |,V76^*|) |),V70+|; |,V25,
V26,V30,|(|V44|) |,V50,/|//|V71/|// |V72/|// |V74/
"KEYWORDS: ",v69(10,10)/##fi

<PgDn> - More <F1>-Guide <F2>-Edit <F3> - Exit to IR
----+----1----+----2----+----3----+----4----+----5---
 MDL,V12,V24,,(|(|V76^Z|: |,V76^*|) |),V70+|; |,V25,
 V26,V30,|(|V44|) |,V50,/|//|V71/|// |V72/|// |V74/
 "KEYWORDS: ",v69(10,10)/##fi

----+----1----+----2----+----3----+----4----+----5---
Error code: 99
--------------
Error message:
--------------
    Unknown command (e.g. misspelled function name
    or command); may also be due to a missing
    closing delimiter

          <ENTER> - More     <F4>-Display format
```

Figure 5-7: MOCDS Editing window

Once editing is over the user may press the <ENTER> key, and MOCDS program will automatically store the corrected version in the corresponding PFT file and display the same at the display/print window for visual verification.

For more information about MOCDS help messages and MOCDS program source code refer to annex 5-2 and 5-3 respectively to this chapter.

ANNEX 5-1: MicroIsis format error codes

Error code	Error message
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 operand of an operator are of different types (e.g. trying to add a string to a number)
28	The first argument supplied to the REF function is not a numerical expression
51	Too many literal 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 8000 characters without EMM and 32767 characters with EMM
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 arguments supplied to 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)

ANNEX 5-2: Help messages

Error code: 1

Error message:

End of format detected while processing a repeatable group. Probably the closing parenthesis delimiting the group is missing

Cause:

Either missing of the parenthesis delimiting the repeatable group or presence of an unconditional literal within a repeatable group

Solution:

Use simple formatting statements. For example:

Given record contents: 24 Neelameghan, A.
24 Chowdhury, G.G.

Format	Output
v24+|;|	Neelameghan, A.; Chowdhury, G.G.

Error code: 2

Error message:

Nested repeatable group (i.e. repeatable group inside a repeatable group.)

Cause:

Probably the display format consists of nested repeatable group

Solution:

Check your display format and make sure that no repeatable group has been nested within another repeatable group

Error code: 8

Error message:

IF command without THEN

Cause:

The keyword "THEN" is missing in one of the IF statements used

Solution:

Check your display format to see in which IF command the keyword THEN is missing, then correct it accordingly by placing the keyword THEN at its appropriate position

General format of the IF command:

IF condition THEN format

Example:

Incorrect format: if p(v1) and a(v2) fi
Correct format : if p(v1) and a(v2) then v3 fi

Error code: 19

Error message:

Unmatched open bracket "("

Notes:

Parentheses are useful for controlling the order of evaluation. Expressions enclosed in parentheses are evaluated first and inner parenthetical expressions are evaluated before outer expressions. For good results number of open brackets must equal number of closed brackets.

Cause:

Probably the open bracket is missing, i.e. the number of open brackets and closed brackets are not equal.

Solution:

Check for missing open bracket in your display format.

Example:

Incorrect format: ref(val(v4)-v3
Correct format : ref(val(v4)-v3)

Error code: 20

Error message:

Unmatched close bracket ")". It may also be caused by an invalid operand in an expression.

Cause:

- Missing close bracket(s); and
- Use of invalid operand in an expression.

Solution:

- Check the display format for the absence of close bracket(s). This means that the number of open brackets must equal to the number of close brackets. Then make appropriate corrections;
- Numerical expressions should be formed using operand which have numerical value and operand specifying the evaluation to be performed.

  Examples of numerical operand are:

  - Numerical constants, e.g. 10, 19.45, 0.35, etc.
  - Numerical functions, e.g. val(v10)
  - MFN, the value of MFN of a record
  - Numerical expressions, e.g. (val(v20)-15)
- String expressions are formed using operand which are string of characters. A string expression always consists of a single operand which may be one of the following:
- Unconditional literal: e.g. 'Journal articles'
- Field selector: which may include an Offset/length command, e.g. v26^c*2.2)
- String functions: e.g. S(mdl,v24,v25,v26):'Technology'

Error code: 26

Error message:

The two operand of an operator are of different type (e.g. trying to add a string operand to a number.

Cause:

Adding a string value to a numeric value and/or vice versa

Solution:

Check your display format to see whether correct operand have been used

Examples:
- Incorrect format: '100' + 10

Note here the value hundred is considered to be a string whereas value 10 is numeric.

- Correct format: 100 + 10
  Here both 100 and 10 are numerical values

Error code: 28

Error message:

The first argument supplied to the REF function is not a numerical expression.

Cause:

Probably a string expression has been supplied to the REF function.

Solution:

Check all statements having used REF function to see whether the first argument is numerical or not.

Example:
Incorrect format: ref('water',v5)
Correct format : ref(val(v4),v5)

Error code: 51

Error message:

Too many literal and/or conditional commands associated with a field selector.

Cause:

Probably the number of delimiters enclosing the literal do not match or more than one which is not allowed by MicroIsis formatting language.

Solution:

- Check the display format to see if delimiters have been used more than necessary. The following literal have to be checked:
- Unconditional literal (')
- Conditional literal (")
- Repeatable literal (¦).

Error code: 53

Error message:

IF command not terminated by FI

Cause:

The keyword FI is missing in one or more of the used IF commands.

Solution:

Check all the IF statements in your display format and see which one or all those not terminated by FI. Remember each of the IF statement must be terminated by FI. Hence, the number of IF statements should equal the number of FIs.

Example:
- Syntax: IF command THEN format FI
- if a(v100) then v200 fi

Error code: 54

Error message:

Plus (+) sign out of context: MicroIsis was expecting a repeatable literal following the plus (+) sign.

Cause:

Using the plus (+) sign with a non-repeatable group.

Solution:

Check the occurrences of (+) to see whether they have been correctly used or otherwise make the appropriate corrections.

Examples:
- Non-repeatable field (v100)
- Incorrect format: "Title"/v100(3,3)+|;|
- Repeatable group (v110)
- Correct format : "Author"/v110(3,3)|;|

Error code: 55

Error message:

Unmatched FI

Cause:

The keyword FI has been used in the format without prior use of IF command.

Solution:

FI must be used as long as IF command is used. The keyword should not be used for other purposes as the MicroIsis will take it to be as part of the formatting language.

Error code: 56

Error message:

Work area overflow: your format produced two much output which MicroIsis cannot handle. The work area size is limited to 8000 characters without EMM and

32767 characters with EMM.

Solution:

It is important to have several worksheets for a given FDT in case of a large database. This will solve the problem of overflow as several display formats will be created for each worksheet.

Error code: 58

Error message:

One or more arguments supplied to the F function are not numerical expressions.

Cause:

- General format of the F function: F(expr-1,expr-2,expr-3)
- All three numerical expressions.
- Probably one or more of the arguments supplied to the F function are not numerical expressions.

Solution:

Check the way arguments have been supplied to the F function and make appropriate corrections.

Example:

- Incorrect format: f('author',1,0)
- Correct format : f(mfn,1,0)

Error code: 60

Error message:

A non string function used as command (only string functions may be used as commands)

Cause:

Non string function has been used as command in the display/print format

Solution:

Only string functions may be used as commands in the formatting language.

Error code: 61

Error message:

The arguments supplied to A or P function is not a field selector

Cause:

A non field selector has been used with A or P function

Note: Field selectors are either fields or sub-fields.

Solution:

Check the display/print format to ensure that "A" or "P" function is being used with field selectors only

Example:

Valid statements: 1. p(v24) 2. p(v24^a)
3. a(v24) 4. a(v24^a)

Error code: 99

Error message:

Unknown command (e.g. misspelled function name or command); may also be due to a missing closing delimiter

Cause:

The command you entered does not conform to the MicroIsis formatting language rules. May be due to:

- Presence of unknown character(s), e.g. *, ;,>, 8, etc. in your display format
- Misspelled name or command
- Presence of incomplete repeatable literal, e.g. |v24^a
- Missing open bracket or both open and close brackets

Solution:

Check for the presence of unknown characters in the display format and spellings of different default names and commands.

Error code: 101

Error message:

Stack overflow (probably due to a too complex expression)

Cause:

Probably pilling together different formatting statements without separating them with commas or spaces.

Solution:

Separate different commands with a comma or space. Check your display format to ensure that this requirement has been met.

Error code: 102

Error message:

Stack underflow (it may be due to unmatched (. If your format is correct it indicates a CDS-ISIS software problem)

Cause:

May be due to the presence of unmatched "(" or otherwise software problem

Solution:

Check for the "(" to see whether it is missing in one or more statements

ANNEX 5-3: MOCDS PROGRAM SOURCE CODE

```
program MOCDS(m: string) [menu];

var nr,mfn,cnt,rc,codemsg,sc,kk,n,row : real;
    dopt,bopt,c,num,lin, code,mm,mrl  : string;

procedure EditFmt;
var i,j,k,no              : real;
    userfmt,userdbn,str3: string;
    str1:array[1..180] of string;
    str2:array[1..30] of string;
begin
  page(3);
  clearbox(1,1,8,80,1);
  box(1,1,8,80,1);
  box(2,50,6,29,2);
  cursor(2,58); write('USER'S GUIDE');
  chattr(4,2,58,12);
  cursor(3,52); write('<F1>    - Consult guide');
  cursor(4,52); write('<F2>    - Edit format');
  cursor(5,52); write('<F3>    - Exit to IR menu');
  cursor(6,52); write('<PgDn> - For more');
  chattr(2,3,53,2);
  chattr(2,4,53,2);
  chattr(2,5,53,2);
  chattr(2,6,53,4);
  clearbox(9,1,6,80,1);
  cursor(9,1);
  write('----+----1----+----2----+----3----+----4');
  write('----+----5----+----6----+----7----+----');
  cursor(14,1);
  write('----+----1----+----2----+----3----+----4');
  write('----+----5----+----6----+----7----+----');
  userfmt:=fmtname;
  cursor(1,61); write('Format name: ',userfmt);
  userfmt:=userfmt|'.PFT';
  assign('inp',path('sys',5)|userfmt);
  i:=1;
     while not (eof(inp)) do
           begin
             readln(inp,lin);
             str1[i]:=substr(lin,1,40);
             str1[i+1]:=substr(lin,41,40);
             i:=i+2;
           end;
           no:=i;
       assign('inp','');
      clearbox(2,2,6,40,1); j:=1; str3:='';
      repeat
        cursor(j+1,3);
        write(str1[j]);
```

```
          str3:=str3|str1[j];
          j:=j+1;
        until (j>6);
        cursor(2,3);
        savescr(2);
        repeat
          sc:=kbdkey(c); uc(c);
          c:=chr(0);
        until (sc=59) or (sc=60) or (sc=61) or (sc=81);
      assign('inp','');
        case sc of
        59: begin
              clearbox(10,1,4,80,1);
              page(2);
            end;
        60: begin
              cursor(13,60); write('Press <ENTER> to Save ');
              chattr(4,13,67,5);
              kk:=edit(str3,255,10,2,240,1,' ');
              if kk=0 then
                 begin
                   assign('out',path('sys',5)|userfmt);
                   writeln(out,str3);
                   writeln(out);
                   EditFmt;
                 end;
            end;
        61: begin
              page(1);
              clear;
              mm:='61';
            end;
        81: begin
              clearbox(2,2,6,41,1);
              j:=1; cursor(j+1,3);
            end;
        end;
 end;

procedure ShowCode;
var row,sc : real;
    lin    : string;

begin
  repeat
    clearbox(15,1,9,80,1);
    box(15,1,9,80,2);
    row:=16;
      repeat
        while not (eof(inp)) and (row < 23) do
             begin
               readln(inp,lin);
```

```
                cursor(row,3);
                writeln(lin);
                row:=row + 1;
                box(15,1,9,80,2);
              end;
              if eof(inp) then
                 begin
                   cursor(24,39); clearLn;
                   write('End of text  <F4> - Display format ');
                   chattr(4,24,39,11);
                   chattr(2,24,54,2);
                   cursor(24,74);
                 end;
                 if not (eof(inp)) then
                  begin
                    cursor(24,39);
                    write('<ENTER> - More  <F4> - Display format ');
                    chattr(2,24,40,5);
                    chattr(2,24,56,2);
                    cursor(24,76);
                  end;
              savescr(3);
              sc:=kbdkey(c);
            uc(c);
      until (sc=28.00) or (sc=62.00);
   until (sc=62.00) or (eof(inp));
 assign('inp','');
 if (sc=62.00) then EditFmt;
end;

procedure ShowMsg;

begin
  code := encint(codemsg,0);
  if code<>''then
  begin
    case code of
     '1'  : begin
              assign('inp', path('sys',5)|'ER_1');
              ShowCode;
            end;
     '2'  : begin
              assign('inp', path('sys',5)|'ER_2');
              ShowCode;
            end;
     '8'  : begin
              assign('inp', path('sys',5)|'ER_8');
              ShowCode;
            end;
     '19' : begin
              assign('inp', path('sys',5)|'ER_19');
              ShowCode;
```

```
            end;
        '20' : begin
                 assign('inp', path('sys',5)|'ER_20');
                 ShowCode;
               end;
        '26' : begin
                 assign('inp', path('sys',5)|'ER_26');
                 ShowCode;
               end;
        '28' : begin
                 assign('inp', path('sys',5)|'ER_28');
                 ShowCode;
               end;
        '51' : begin
                 assign('inp', path('sys',5)|'ER_51');
                 ShowCode;
               end;
        '53' : begin
                 assign('inp', path('sys',5)|'ER_53');
                 ShowCode;
               end;
        '54' : begin
                 assign('inp', path('sys',5)|'ER_54');
                 ShowCode;
               end;
        '56' : begin
                 assign('inp', path('sys',5)|'ER_56');
                 ShowCode;
               end;
        '60' : begin
                 assign('inp', path('sys',5)|'ER_60');
                 ShowCode;
               end;
        '61' : begin
                 assign('inp', path('sys',5)|'ER_61');
                 ShowCode;
               end;
        '99' : begin
                 assign('inp', path('sys',5)|'ER_99');
                 ShowCode;
               end;
       '101': begin
                 assign('inp', path('sys',5)|'ER_101');
                 ShowCode;
               end;
       '102': begin
                 assign('inp', path('sys',5)|'ER_102');
                 ShowCode;
              end;
    end;
  end;
end;
```

```
Function FindCode(r: real): real;
var rc,codevalue : real;
begin
  savescr(1);
  FindCode:=0;
  rc:=record(r);
  codevalue:=format(78);
  codemsg:= codevalue;
  EditFmt;
  ShowMsg;
end;
begin   { === main program === }
  dopt:='D'; bopt:='B';
  if m=dopt then
     begin
       nr:=setpos(0,0);
       cnt:=1;
       mfn:=setpos(0,1);
     end
     else
       begin
         clear;
         mfn:=1;
         cnt:=mfn;
         nr:=maxmfn-1;
       end;
         clearbox(1,1,25,80,1);
         c:=chr(0);  mm:=encint(sc,0); mrl:='';
         while ((mfn>0) and (c=chr(0)) and (mm<>'61')) do
         begin
           rc:=FindCode(mfn);
         end;
           if rc=0 then
              begin
                cnt:=cnt+1;
                if m=dopt then mfn:=nxtpos(1)
                else
                  if mfn=nr then mfn:=0
                  else
                    mfn:=mfn+1;
              end;
     if (mm='61')  then mrl:=mrl|'?S';
 autotype(mrl);
 m:=' ';
end.
```

CHAPTER 6

SUMMARY AND RECOMMENDATIONS

6.1 SUMMARY

The main objective of this study is to improve the quality of the information services by ensuring some aspects of the quality of input data entered into the databases created using MicroIsis software package. The study reviews the historical background of the software, development perspectives and its general features. The CDS-ISIS Pascal library associated with the software is also reviewed. The flexibility of the software in terms of export/import of data, availability of powerful indexing techniques, formatting and search language, local area network support, the development of the software being supported by an international intergovernmental organization, Unesco, its usefulness in information storage and retrieval system (ISRS) and above all being provided free of cost to nonprofit making organizations is gaining its wide and fast acceptance world-wide.

Reasons for the occurrence of errors at the input stage and efforts made by different DBMS and text retrieval software developers to minimize errors during

data entry stage by incorporating some house-keeping routines designed to provide general consistency checks are reviewed. Some of the attempts at improving the quality of input data entered into the databases created using MicroIsis software package are mentioned. For example, to minimize typing errors and save time, the standard MicroIsis has facility for entering default data values in one or more fields of a database either permanently or for temporary use.

The study indicates that it is now feasible given the developments in microelectronics technology by which microcomputers of high speed and high storage capacity are now widely available, to access database guides online rather than the traditional approach of having them only in printed forms and consulted manually.

The Data Entry Assistance (DEA) program is developed to assist MicroIsis users at two levels:

- To provide online guidance to data entry operators on how to enter data elements in different fields in a record of database created using MicroIsis software package, in particular those databases developed in conformity to ABNCD record structures; and
- It is also developed to provide guidance on authority list(s). It enables selecting data from appropriate authority list(s) or vocabulary

contorl tools and add the selected data to specific field(s) in one or more records in a database.

The MOCDS program is developed to provide online help on format errors when attempting to display records by using option B 'Browse' or D 'Display search results' in information retrival services. The program provides information relating to the type of error, a brief message explaining the type of error invoked, cause and what should be done to rectify the error.

6.2 RECOMMENDATIONS

The programs have been tested by some MicroIsis users and based on the feedback provided some modifications have been effected. The MOCDS program is capable of displaying and editing three lines at a time. This limitation is due to the fact that, the CDS-ISIS Pascal field editor allows to edit a maximum of 255 characters at a time. This is cumbersome and time consuming especially when editing a large display format of say more than one page.

MOCDS program can optimally be used if integrated with the standard MicroIsis source code in which the display module can be retained and displayed at a window on the same user display/print format. This will enable

use of the 21 lines reserved for data in the MicroIsis editing screen. This study was designed specifically to improve the quality of the information services by ensuring some aspects of the quality of input data.

Similar developments can be initiated aimed at improving the MicroIsis software especially in the following areas:

- Online guidance is needed in the preparation of display/print format to guide users on how to use different formatting statements; and
- A brief online help explaining the function(s) of each of the functions provided by MicroIsis main menu. For example a naive user may not know what the 'Master file services' function does and so on.

This type of help facility will encourage and promote the application of MicroIsis software package.

BIBLIOGRAPHY

1. Abebe Rorissa et al. 1992. ABNCD+: A prototype for an integrated information storage and retrieval system. Microcomputer for Information Management. 9(3): 137-159.

2. Atherton, P. 1978. Standard for user-system interface language in online retrieval systems: The challenge and the responsibility. Online Review 2. 1978: 57-61.

3. Bamford, Carl 1987. Data structures, files and databases. Houndmills: Macmillan Education Ltd.

4. Bawden, David and Karen Blackman 1990. IT strategies for information management. London: Butterworth & Co.

5. Birru Dori 1992. Development of Micro-CSD-ISIS Amharic version. MSc.I.S. thesis. SISA, AAU: Ethiopia.

6. Chabris, Christopher F. 1989. Artificial intelligence and Turbo Pascal. New Delhi: Galgotis Publications Pvt. Ltd.

7. Chaudhry, Ali Anwar 1989. Interfacing CDS-ISIS with Wordstar: ICIMODS's experience. Program 23(1): 65-72.

8. Cho, C-K. 1990. An introduction to software quality control. New York: John Wiley & Sons.

9. Chorafas, D. N. 1986. Fourth and fifth generation programming languages. Vol. 1: integrated software database languages and expert systems. New York: McGraw-Hill Book Company.

10. Chowdhury, G.G. et al. 1994. Vocabulary control online in MicroIsis databases: a pascal interface. (Unpublished).

11. Cortez, Edwin M. and Kazlanskas, Edward John. 1986. Managing information systems and technologies. New York: Neal-Schuman Publisher, Inc.

12. Davies, Roy (ed). 1986. Intelligent information systems: Progress and prospects. Chichester: Ellis Horwood Limited.

13. Davis, Gordon B. Computer data processing. 2nd edition. London: McGraw-Hill Book Company.

14. Davis, Gordon B. and Margrethe H. Olson 1985. Management information systems: conceptual foundations, structure and development. 2nd edition. New York: McGraw-Hill Book Company.

15. Debons, Anthony 1988. Information service: an integrated view. Beston: G. K. Hall & Co.

16. ECA, PADIS. The use of microcomputers in documentation centres in Africa. Paper presented in the Fifth Session of the Joint Conference of African Planners, Statisticians and Demographers held at Addis Ababa, 15-20 March, 1988.

17. ECA, PADIS. The use of microcomputers in documentation centres in Africa. Paper presented in the Sixth Session of the Joint Conference of African Planners, Statisticians and Demographers held at Addis Ababa, 21-28 March, 1990.

18. Ganzmann, Jochen 1990. Criteria for the evaluation of thesaurus software. International Classification 17(3-4): 148-157.

19. Gerrie, B. 1983. Online information systems: use and operating characteristics, limitations, and design alternatives. Arlington: Information Resource Press.

20. Haneock-Beaulieu, M. 1992. User friendliness and human computer interaction in online library catalogue. Program 26(1): 36-37.

21. Konvalina, John and Wileman, Stanley 1987. Programming with Pascal. New York: McGraw-Hill Book Company Inc.

22. Kumar, Girja and Kumar, Krishan 1975. Theory of cataloguing. New Delhi: Vikas Publishing House Ltd.

23. Lancaster, F.W. 1979. Information retrieval systems: characteristics, testing and evaluation. New York: John Wiley & Sons.

24. Lancaster, F.W. 1977. The measurement and evaluation of library sciences. Arlington: Information Resource Press.

25. Lauro, Anne Di 1990. Manual for preparing records in microcomputer based bibliographic information systems. Ottawa: IDRC.

26. Lopes, Maria Ines 1993. Subject authority control in a union catalogue: perspectives for a rational co-operative project in Portugal. Knowledge organisation 20(3): 133-138.

27. Marchionini, G. 1992. Interface for end user information seeking. Journal of the American Society for Information Science 43(2): 156-163.

28. Molla Hunegnaw 1993. Enhancement of user-friendliness of Micro CDS-ISIS for search and retrieval phase. MSc.I.S. thesis. SISA, AAU: Ethiopia.

29. Molla Hunegnaw and Neelameghan, A. 1993. System Interface Search Assistance for micro CDS-ISIS: In NIT'93. Sixth International Conference on New Information Technologies held at San Juan, Purto Rico, 11-13 Nov. 1993: 153-166.

30. Neelameghan, A. 1992. Home-grown non-bibliographic object oriented databases. In NIT'92. Fifth International Conference on New Information Technologies held at Hongkong, 30 November - 2 December, 1992: 259-284.

31. Nieuwenhuysen, P. 1991 Computerised storage and retrieval of structured text information :CDS-ISIS version 2.3. Program 25(1) 1-18.

32. Ornager, Sussane 1990. CDS-ISIS (Mini-micro version 2.3): supplement to CDS-ISIS reference manual. Bonn: German Foundation for International Development (DSE). Education Science and Documentation Centre.

33. Osborne, D. J. 1985 Computers at work. Chichester: John Willey & Sons.

34. Perera, P. 1992. Micro CDS-ISIS: a critical appraisal of its search interface. Program 26(4): 373-386.

35. Rao, I.K. Ravichandra 1990. Library automation. New Delhi: Wiley Eastern Ltd.

36. Raitt, David I. (ed). 1984. An introduction to online information systems: a collection of the significant papers in the field of the online retrieval of information. Oxford: Learned Information Ltd. 1984: 243-252.

37. Rijsbergen, C. J. van 1990. Information retrieval. 2nd edition. London: Butterworth and Co. Ltd.

38. Rowley, J.E. 1990. The basics of systems analysis and design for information managers. London: Clive Bingley Limited.

39. Simai, M. 1990. Global technology and the world economy in the late twentieth century. London: Printer Publishers Ltd.

40. Simmons, Peter and Alan Hopkinson(eds). 1992. CCF/B: The Common Communication Format for Bibliographical Information. Paris: Unesco.

41. Strilatha, G. and Aparna, S. 1992. A user - interface for information retrieval in CDS-ISIS database. Library Science 29(3): 127-134.

42. Teshome Bula 1993. Computer assissted generation of vocabulary control tools and their application to databases in urban planning. MSc Thesis. SISA Addis Ababa University, Addis Ababa: Ethiopia.

43. Unesco 1989. Mini-micro CDS-ISIS (Reference Manual version 2.3). Paris: Unesco.

44. Woods, Lawrence A. and Pope, Nolan F. 1983. The librarian's guide to microcomputer technology and applications. White Plains: American Society for Information Science. 1983: 33-44.