

BIREME / PAHO / WHO

Latin American and Caribbean Center on Health Sciences Information

xisis Platform - Administration Manual

Version 1.1 "beta"

São Paulo - 2005

Copyright © 2005 - BIREME / PAHO / WHO

xisis Platform - Administration Manual

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Card catalog

BIREME / PAHO / WHO (Brazil)

xisis Platform - Administration Manual. / BIREME (org.).

São Paulo : BIREME / PAHO / WHO, 2005.

28 p.

1. User manual. 2. Information access. 3. Information systems. 4. Information management. 5. Public health. 6. Public Health services. I. BIREME II. Title

Warning - Any mention in this document to companies, institutions, persons or products are not an endorsement or recommendation given by BIREME / PAHO / WHO, thus it does not mean a preference to a similar one, cited or not.

BIREME / PAHO / WHO

Latin American and Caribbean Center on Health Sciences Information

Rua Botucatu, 862 - V. Clementino

This document was produced with the Documents Conformation Methodology (NorDoc) developed by BIREME.

Table of contents

Abbreviations used	II
How to use this manual	IV
Preface	1
About BIREME	1
The Virtual Health Library (VHL)	2
Introduction	5
Managing collections	6
Creating collections	6
Deleting collections.....	8
Listing existing collections.....	8
Indexing or re-indexing collections.....	9
Obtaining information of a collection status	10
Unlocking a collection.....	11
Server administration	12
Server configuration	12
Starting the server	13
Ending server operation.....	13
Data backup	14
Making a backup of the server	14
Recovering a backup in the server	14
Exporting and importing collections	16
Exporting a collection	16
Importing data to a collection	17
Index/Reindex the collection	17
Further information	18
Bibliographic references	19
Glossary	20

Abbreviations used

- ANSI. American National Standards Institute.
- ASCII. American Standard Code for Information Interchange.
- BIREME. Latin American and Caribbean Center on Health Sciences Information.
- BVS. Biblioteca Virtual em Saúde (*see* VHL).
- CGI. Common Gateway Interface.
- DTD. Document Type Definition.
- FAPESP. Fundação de Amparo à Pesquisa do Estado de São Paulo [The State of São Paulo Research Foundation].
- FST. Field Selection Table.
- HTML. HyperText Markup Language.
- HTTP. HyperText Transfer Protocol.

- ISO. International Organization for Standardization.
- PAHO. Pan American Health Organization.
- PDF. Portable Document Format.
- SGML. Standard Generalized Markup Language.
- UMLS. Unified Medical Language System.
- UNESCO. United Nations Educational, Scientific and Cultural Organization.
- URL. Universal Resource Locator.
- VHL. Virtual Health Library.
- WHO. World Health Organization.
- XHTML. eXtensible HyperText Markup Language.
- XML. eXtensible Markup Language.
- XSL. eXtensible Stylesheet Language.
- XSLT. eXtensible Stylesheet Language Transformations.

How to use this manual

This document is organized in four chapters comprising the setup procedures and the use of XISIS as a server and ISIS database manager containing data as XML (collections).

Therefore it is not a tutorial (step by step) nor a quick reference, but an essential documentation to the XISIS functions and the data type it can manage.

Preface

About BIREME

Year after year, BIREME has been following its mission of being a center dedicated to scientific and technical health information for the region of Latin America and the Caribbean. Founded in Brazil in 1967, under the name of Regional Medicine Library (which the acronym BIREME comes from), it has always met the growing demand for up-to-date scientific literature from the Brazilian health systems and the communities of healthcare researchers, professionals and students. Then, in 1982, its name changed to Latin-American and Caribbean Center on Health Sciences Information so as to better express its dedication to the strengthening and expansion of the flow of scientific and technical health information across the region, but kept the acronym.

Networking, based on decentralization, on the development of local capacities, on sharing information resources, on developing cooperative products and services, on designing common methodologies, has always been the foundation of BIREME's technical cooperation work. It has been like this that the center established itself as an international model that fosters professional education with managerial and technical information with the adoption of information and communication paradigms that best meet local needs.

The main foundations that gave origin and which support the existence of BIREME are following:

- ✓ access to scientific and technical health information is essential for the development of health;
- ✓ the need to develop the capacity of Latin American and Caribbean countries to operate their sources of scientific-technical health information in a cooperative and efficient manner;
- ✓ the need to foster the use and to respond to the demands for scientific-technical health information from governments, health systems, educational and research institutions.

BIREME, as a specialized center of the Pan-American Health Organization (PAHO)/ World Health Organization (WHO), coordinates and conducts technical cooperation activities on the management of scientific information and knowledge with the aim of strengthening and expanding the flow of scientific health information in Brazil and in other Latin American and Caribbean countries as a key condition for the development of health, including its planning, management, promotion, research, education, and care.

The agreement that supports BIREME is renewed every five years by the members of the National Advisory Committee of the institution (PAHO, Brazilian Ministry of Health, Brazilian Ministry of Education and Culture, Secretary of Health of the State of São Paulo, and Federal University of São Paulo – Unifesp). The latter provides the physical infrastructure necessary for the establishment of the institution.

In 2004 the institution took on the responsibility of becoming a knowledge-based institution.

The Virtual Health Library (VHL)

With the rise and consolidation of the internet as the prevailing means of access to information and communication, BIREME's technical cooperation model evolved,

as of 1998, to build and develop the Virtual Health Library (VHL) as a common space for the convergence of the cooperative work of producers, intermediaries, and users of information. The VHL promotes the development of a network of sources of scientific and technical information with universal access on the internet. For the first time there has been a real possibility of equal access to health information.

To BIREME, the Virtual Health Library is a model for the management of information and knowledge, which includes the cooperation and convergence between institutions, systems, networks, and initiatives of producers, intermediaries, and users in the operation of networks of local, national, regional and international information sources favoring open and universal access.

Today, every country in Latin America and the Caribbean (Region) participates either directly or indirectly in the cooperative products and services offered by the VHL, which includes over 1,000 institutions in more than 30 countries.

The VHL is simulated in a virtual space of the internet formed by a collection or network of health information sources in the Region. Users of different levels and locations can interact and navigate in the space of one or many information sources, regardless of where they are. Information sources are generated, updated, stored and operated on the internet by producers, integrators, and intermediaries, in a decentralized manner, following common methodologies for their integration in the VHL.

The VHL organizes information in a structure that integrates and interconnects reference databases, specialist directories, events and institutions, a catalogue of the information resources available on the internet, collections of full texts with a highlight for the SciELO (*Scientific Electronic Library Online*) collection of scientific journals, selective information dissemination services, information sources to support education and decision-making, news, discussion lists, and support to virtual communities. The space of the VHL is, therefore, a dynamic and decentralized network of information sources based on which it is possible to retrieve and extract information and knowledge to support health decision-making processes.

The Virtual Health Library can be visualized as a distributed base of scientific and technical health knowledge that is saved, organized and stored in electronic format in the countries of the Region, universally accessible on the internet and compatible with international databases.

Introduction

Xisis is a database server that stores documents in XML. Documents are stored in collections and may be consulted by their structure.

Differently from other initiatives in the scope of XML databases, *xisis* proposes a model in which all operations in the server are performed through calls in XML.

Xisis is developed in Java and inherits concepts of ISIS databases for its indexation. Currently, the index generation process uses the C language code of the *Isis_dll* function library developed by BIREME/OPS/OMS. We hope to implement this functionality in Java so as to ensure its operation regardless of the platform.

Managing collections

The function called *xisis_control* is used to administrate xisis, and possible operations are:

- Creating collections
- Deleting collections
- Listing existing collections
- Indexing or re-indexing collections
- Obtaining information of a collection status
- Unlocking a collection

Communication with the server is performed through messages in XML. These messages can be sent from the application that uses the functions internally or with the demonstration application available when you install the server.

Creating collections

Collection creation is performed through the function *xisis_control*, sending parameters to create a new collection (*new*).

```
<parameters>
  <collection></collection>
  <new>
    <indexing>
      <index name="">
        <extract path="" technic="words"></extract>

```

```

    </index>
    <stopwords>
      <word></word>
    </stopwords>
  </indexing>
</new>
<output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>

```

<collection>

Name of the collection in which the operation will be performed. When you create a new collection, we recommend you include a subdirectory, for example “my_collection/my_collection”. A collection is a set of files that are stored in a server directory. If a subdirectory is not indicated, the collection will be stored directly in the directory configured (see the configuration section).

<new>

Indicates that the collection will be created

<indexing>

Contains the information for the creation of collection indexes.

<index name="">

Repetitive element that contains the data for each collection index. The attribute “name” indicates the index name.

<extract path="" technic="">

Repetitive element that contains the parameters for the extraction of data used in index generation. The attribute “path” indicates the path in the document to obtain the index content. The attribute “technic” defines the key extraction technique; possible techniques include: “words” to obtain each word of the content or “key” to consider all the content with a maximum length of 60 characters.

<stopwords>

List of words that are not considered in index generation.

<word>

Repetitive element that contains each of the words *stopword*. Note: in this version, the word list should be in alphabetical order and in capital letters.

<output encoding="" omit-xml-declaration="">

Indicates how to handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Deleting collections

Collection deletion is performed through the function *xsis_control*, sending parameters to delete the collection (*delete*).

```
<parameters>
  <collection></collection>
  <delete/>
  <output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>
```

<collection>

name of the collection in which the operation will be performed.

<delete>

Indicates that the collection will be deleted.

<output encoding="" omit-xml-declaration="">

Indicates how you will handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Listing existing collections

You can consult the existing collections in the server through the function *xsis_control*, sending parameters to list the collections (*collectionList*).

```
<parameters>
  <show>
    <collectionList/>
  </show>
  <output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>
```

<show>

Has the parameters of what is going to be shown as the result of the operation.

<collectionList>

Indicates that a list of the collections available in the server should be produced.

<output encoding="" omit-xml-declaration="">

Indicates how you will handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Indexing or re-indexing collections

To index or re-index a collection you should indicate suitable parameters for the function *xsis_control* (*reindex*)

```
<parameters>
  <collection></collection>
  <reindex>
    <indexing>
      <index name="">
        <extract path="" technic="words"></extract>
      </index>
      <stopwords>
        <word></word>
      </stopwords>
    </indexing>
  </reindex>
  <output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>
```

<collection>

Name of the collection in which the operation will be performed.

<reindex>

Indicates that the operation to be performed is to re-index the collection.

<indexing>

Contains the block of indexes that is defined for the collection.

<index name="">

Repetitive element that contains the data for each collection index. The attribute “name” indicates the index name.

<extract path="" technic="">

Repetitive element that contains the parameters for the extraction of data used in index generation. The attribute “path” indicates the path in the document to obtain the index content. The attribute “technic” defines the key extraction technique; possible techniques include: “words” to obtain each word of the content or “key” to consider all the content with a maximum length of 60 characters.

<stopwords>

List of words that are not considered in index generation.

<word>

Repetitive element that contains each of the words *stopword*. Note: in this version, the word list should be in alphabetical order and in capital letters.

<output encoding="" omit-xml-declaration="">

Indicates how you will handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Obtaining information of a collection status

The server xsis enables you to consult the general status of a collection through the control function (*xsis_control*) through the element *show*. You can request information on the quantity of active documents, document lockup –with the parameter *status*- and the indexes available (*indexing*).

```
<parameters>
<collection></collection>
<show>
  <status/>
  <indexing/>
</show>
<output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>
```

<collection>

Name of the collection in which the operation will be performed.

<show>

Has the parameters that should be shown as a result of the operation.

<status>

Indicates that you expect information about the collection status: quantity of active documents (not erased) and document lockup status.

<indexing>

Contains the information for the creation of indexes of the collection.

```
<output encoding="" omit-xml-declaration="">
```

Indicates how you will handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Unlocking a collection

Due to operations that are not finalized because of external reasons (power outage, for example) sometimes the documents of a collection may be in a lockup status although there is not any operation in execution. The function *xsis_control* may receive the parameter *unlock*.

```
<parameters>
  <collection></collection>
  <unlock>
    <full/>
    <document id=""/>
  </unlock>
  <output encoding="iso-8859-1" omit-xml-declaration="no"/>
</parameters>
```

```
<collection>
```

Name of the collection in which the operation will be performed.

```
<unlock>
```

Indicates that the operation to perform is unlock.

```
<full>
```

Informs that the operation should be applied to all documents of the collection.

```
<document id="">
```

Repetitive element that informs to which documents the operation should be applied.

```
<output encoding="" omit-xml-declaration="">
```

Indicates how you will handle the XML declaration. The standard result returns with ISO-8859-1 encoding; the option “encoding” enables you to request other options. The attribute “omit-xml-declaration” enables you to delete the declaration in the return, which is useful when you wish to compose XML files as a sequence of nodes of different sources.

Server administration

Server configuration

There are two possible server configurations: define the xisis service *port* (the predetermined option is 8800) and the place where the collections will be stored.

Adjusting the service port

The adjustment of a service port is performed in the system start file XI_Start_JettyServer.bat in Windows or XI_Start_JettyServer.sh in Linux. These files are available in *<path_to_xisis>/xisis/bin*.

Edit the file and change it to the new value desired.

```
@%JAVA_HOME%\bin\java -server -Djava.library.path=..\lib -cp
..\lib\xisis.jar;..\lib\ISISJAVA.jar;..\lib\org.mortbay.jetty.jar;..\lib\javax.servlet.jar;..\
lib\xmlrpc-1.1.jar;..\lib\dom4j.jar;..\lib\zeus.jar xisis.XI__JettyServer ../html 8800
```

Defining the collections directory

The collections comprise a set of files and are stored in the directory indicated in the file *<path_to_xisis>/xisis/bin/xisis.cfg*. The *path* indicated may be relative or absolute. The predetermined directory is in *<path_to_xisis>/xisis/*

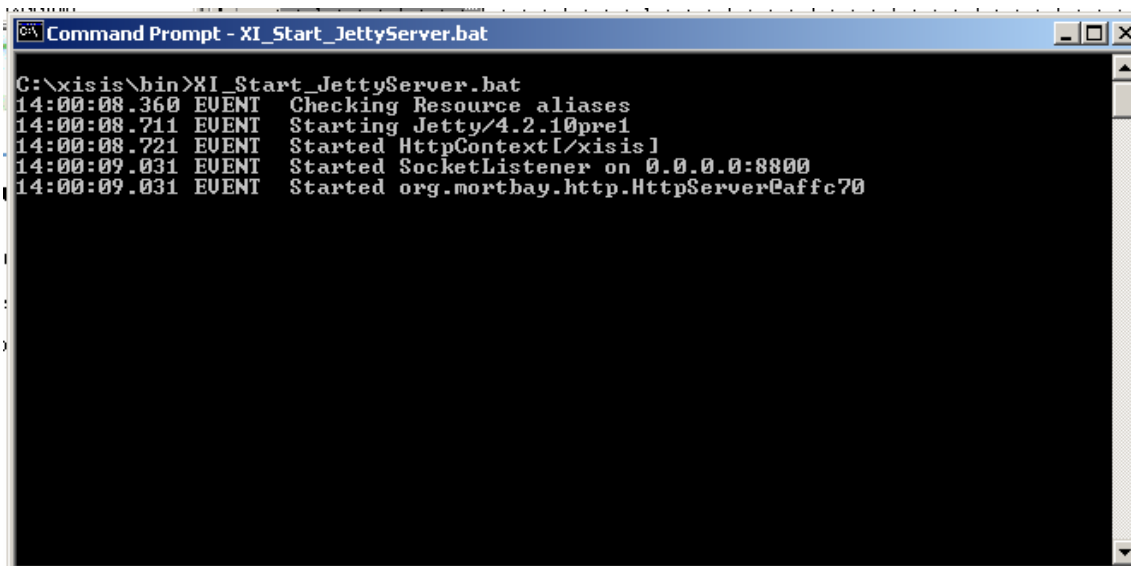
Starting the server

The server should be started in the directory where xsis is installed, specifically in `<path_to_xsis>/xisis/bin`.

Starting the server in Windows

```
C:\>cd <path_to_xsis>\xisis\bin
C:\xisis\bin> XI_Start_JettyServer.bat
```

The result of this operation should present the situation shown in the figure.



```
Command Prompt - XI_Start_JettyServer.bat
C:\xisis\bin>XI_Start_JettyServer.bat
14:00:08.360 EVENT   Checking Resource aliases
14:00:08.711 EVENT   Starting Jetty/4.2.10pre1
14:00:08.721 EVENT   Started HttpContext[/xisis]
14:00:09.031 EVENT   Started SocketListener on 0.0.0.0:8800
14:00:09.031 EVENT   Started org.morthbay.http.HttpServer@aaffc70
```

Starting the server in Linux

The process is the same, but you will use the file `XI_Start_JettyServer.sh`.

```
>cd <path_to_xsis>/xisis/bin
> XI_Start_JettyServer.sh
```

Ending server operation

To end the server operation, all you have to do is enter the command `<Ctrl>+C`.

Data backup

Currently the server data backup consists of simply stopping the server and copying the directory defined in `xisis.cfg`.

Making a backup of the server

Assuming that the collections are stored in the predetermined directory, the steps are:

1. Stop the server
2. Copy the directory `<path_to_xisis>/xisis/collections` to the device desired (another directory, disk unit, etc.).
3. Restart the server

Recovering a backup in the server

The process is simple:

1. Deleting the current content
 - a) Stop the server
 - b) Delete the directory `<path_to_xisis>/xisis/collections`
 - c) Replace it with the directory and backup data
 - d) Restart the server

2. Maintaining new existing collections.

This procedure only works in case there are no collections with an identical name.

- a) Stop the server
- b) Copy the backup data to the directory <path_to_xisis>/xisis/collections
- c) Restart the server.

Exporting and importing collections

There is a set of utility programs to export and import data of a xisis collection. These utility programs should be installed separately. The data are managed in ISO2709 format and contemplate the possibility of importing data from traditional ISIS databases.

Exporting a collection

To export a collection, you will use the utility program **xisisExport**. The exporting parameters are:

```
>xisisExport collection=<name de the collection> iso=<name del file iso> tell=<n>
```

The command *tell* enables you to configure the processing output. For example, `tell=100` will generate a line in the output every 100 documents processed.

The ISO file generated by this tool is called `iso_xisis` to be differentiated from those ISO files that come from traditional ISIS databases.

Importing data to a collection

You can import data to an existing collection from an ISO file coming from a traditional ISIS database or from an ISO file generated in a xsis collection.

```
>xsisImport [iso|iso_xisis]=<file iso> [create|append]=<collection> [fdt=<file>]
<options>
```

xsisImport receives as mandatory parameters an ISO file generated from an ISIS base (iso) or from a collection (iso_xisis). Then, you will indicate whether you wish to create a new collection (create) or add the data to an existing collection (append). If it is an ISO file coming from ISIS, you should indicate the FDT so that the program can transfer the records using elements names (for example, author, date, etc.).

<options> enables you to specify if you wish to perform the ANSI operation of the characters (convert=ansi) and additionally ask to retro-feed the import operation with tell=<n>.

Index/Reindex the collection

Once new data is imported, you should re-index the collection. For this, you should use the utility program xsisIndexing.

```
>xsisIndexing collection=<collection> [reindex=<file>]
```

The option “reindex” is optional and enables you to modify the current index structure.

Further information

For further information about the Xisis Platform and other ISIS family applications can be obtained in the VHL Model Site <http://bvsmodelo.bvsalud.org/>, where you will have access to other manuals.

Bibliographic references

1. UNESCO. *Mini-micro CDS/ISIS: Reference manual (version 2.3)*. Organized by Giampaolo Del Bigio. Paris: United Nations Educational, Scientific and Cultural Organization, 1989. 286 p. ISBN 92-3-102-605-5.
2. BUXTON, Andrew, HOPKINSON, Alan. *The CDS/ISIS for Windows Handbook* [online]. Paris: United Nations Educational, Scientific and Cultural Organization, 2001 [cited 30 August 2006]. 164 p. Available from internet: <<http://bvsmodelo.bvs.br/download/winisis/winisis-handbook-en.pdf>>.
3. SUTER, Tito. "Prehistoria" e historia del MicroISIS [online]. In: *Manual para instructores de Winisis*. Buenos Aires: Centro Atómico Constituyentes (CAC), Comisión Nacional de Energía Atómica (CNEA), 1999 [citado el 30 Agosto 2006]. p. 21-26. Disponible en internet: <<http://www.cnea.gov.ar/cac/ci/isis/isidams.htm>>.

Glossary

- **Application.** Program used to execute tasks in connection with an application, such as the creation or edition of texts, drawings, animations, layout, etc. E.g. : text processor, database manager, Internet browser, etc.
- **Backup.** Procedure used to duplicate one or more files and/or directories in another storing device (tape or disc), thus producing a backup copy that may be restored in the event of accidental deletion or physical damage to the original data.
- **Bibliographic Database.** Electronic version of a catalog or bibliographic index.
- **Browser.** Internet page navigator, such as Internet Explorer and Netscape Navigator.
- **CDS/ISIS - MicroISIS.** Software program developed and maintained by UNESCO to manage bibliographic data.

- **CGI.** The Common Gateway Interface is a standard for interfacing external applications with information servers, such as HTTP or Web servers.
- **Database.** Collection of data that are structured to be easily accessed and handled. It is formed by units called records whose attributes are represented by fields. For example, in a file called "customer base", each customer is a record, with several fields such as "NAME", "CUSTOMER CODE", "TELEPHONE" etc.
- **Electronic Format.** Any form of storage, retrieval or presentation of information that may be transmitted on-line or recorded in magnetic or optical media.
- **Field.** See Database.
- **File.** In computing, a set of data that may be saved into some type of storing device. The data files are created by applications, such as a text processor for example.
- **ISO Format (of files).** Standard established by the ISO to allow the exchange of data between institutions, networks and users.
- **PDF.** File format developed by Adobe whose objective is to maintain the presentation format of a document designed for printing when this document is stored in digital media.
- **SGML.** Metalanguage standard of the ISO (International Organization for Standardization) used for the definition of languages of marking of electronic texts, making possible the interchange and the distribution of documents in the most varied formats, from one same source of data.

- **TCP/IP Protocol.** Standard that defines the method of communication between digital equipment. It employs a single number of identification.
- **URL.** Standard defined for the addressing of data contents via the TCP/IP protocol. Internet browsers use the URL to access Web pages.
- **XML.** Language created to allow the arrangement of data in a structured and hierarchical manner, thus facilitating data communication between different systems and platforms.
- **XSL.** Language created to allow the navigation, selection and capture of data of an XML file.