

IAH USER'S MANUAL

Version 3.1

Table of Contents

Objectives of the chapter	3
Introduction	3
Structure of IAH	4
Area of documents and executable scripts	4
Area of executable scripts	4
Area of databases	4
Default structure of IAH in ABCD	4
Files and directories of the IAH package	5
<i>BASES/</i>	5
<i>HTDOCS/IAH</i>	6
<i>CGI-BIN</i>	6
Understanding the configuration of IAH	7
Contents of the file <i>iah.def.php</i>	7
<i>PATH</i> section	7
<i>APPEARANCE</i> section	8
<i>Style sheet structure</i>	8
<i>HEADER</i> section	8
<i>IAH</i> section	9
How to configure the file <i>iah.def.php</i>	9
Initial testing of the interface	9
Your own databases	10
How to install a new database	10
How to configure a new database	11
Section <i>FILE_LOCATION</i>	11
Section <i>INDEX_DEFINITION</i>	12
Section <i>APPLY_GIZMO</i>	13
Section <i>FORMAT_NAME</i>	14
Section <i>HELP_FORM</i>	14
Section <i>PREFERENCES</i>	15
Management of multiple databases with a different layout	17
How to use the IAH interface	19
Free form	19
<i>Important remarks</i>	19
Basic and Advanced Forms	20
<i>How to search</i>	20
<i>Display of the results</i>	21
Logical operators in the search	23
Search fields	23
<i>Words</i>	23
<i>Descriptor</i>	23
Appendix A - Indexing Techniques	25
Short example of an FST with prefixes for a database compatible with the LILACS format	27
Short example of definitions (<base>.def file) compatible with the FST with prefixes	27

Appendix B - Fields of the LILACS format 27
Appendix C - Searchable fields of the LILACS demo database 29
Appendix D - How to construct IAH URLs with parameters 30
Appendix E - Fields of the interface..... 30

IAH USER'S MANUAL

Objectives of the chapter

This chapter is intended to provide technical information on the installation, setup and maintenance of IAH interface with CDS/ISIS, WinISIS, LILDBI-Web and any compatible databases using ISIS architecture.

A minimum knowledge of CDS/ISIS database standard; solid network, web server and internet expertise as well as full domain of operating system used are the basic requirements to the system and/or support analyst that will use this manual.

An abbreviations list, a glossary of associated terminology and a series of appendices complete information in the manual. At the end, a list of bibliographic references to the methodologies and technologies associated to IAH are available.

Introduction

The Interface for Access to Health Information (IAH) was designed for information retrieval from ISIS databases in an optimum manner over the Internet.

Written in the IsisScript language (the native code of WWWISIS, <http://bvsmodelo.bvsalud.org/php/level.php?lang=en&component=31&item=2>), it was developed for BIREME from 1999, having evolved in conjunction with WWWISIS to support new functions such as the reading of large records (1 Mb), generation of XML content, etc.

IAH can be installed on any PC-compatible that has a web server installed and configured. It has been tested on Internet Information Server (IIS) versions 4, 5 and 6, and on Apache Web Server (Apache) versions 1.3.xx. and 2.x. Apache was tested exhaustively under Windows and Linux.

It has been tested under Apache 2.xx with a minor change. Due to a difference in server global variables' contents, the files named 'index.htm' below htdocs/iah/<lang>/ has to be renamed to iah.htm so the PATH_TRANSLATED variable returns the path properly.

It has also been tested under IIS 6 provided that IIS internal firewall has been set to allow the use of the Common Gateway Interface (CGI) as well the WWWISIS executable. Notice that a Virtual Script directory must be set with permissions to execute scripts and executables.

WWWISIS is the active component of the interface which gives the IsisScript code multiuser access to the ISIS databases across the Common Gateway Interface (CGI).

IAH comes packaged in ABCD with the standard version of WWWISIS, with a key size of 16/60.

The availability of the versions mentioned should be checked with BIREME because this change can bring undesirable effects and other implications for the application.

Structure of IAH

The IAH interface comprises three different parts, in accordance with the following model:

Area of documents and executable scripts

Primary location of the documents (HTML, javascript, CSS, images, configuration files) of the application and the webserver. Normally called DOCUMENT_ROOT.

Location of the executable files or scripts of the application. In this area is located the code of the application, written in the language IsisScript (compiled or not).

Area of executable scripts

Location of the executable programs. In this area is located the WWWISIS program (wxis.exe). Normally called CGI-BIN, this area uses the methods and protocols defined for the CGI interface. Its name can vary according to the web server.

Area of databases

Location of the databases used by the application.

Table 1 shows the the correlation between the above areas and the folder names in various webservers:

Area	Server	
	Apache	IIS
Documents	htdocs	wwwroot
CGI	cgi-bin	scripts
Databases	bases	bases

Table 1: areas for the webserver.

Note: The database directory can be created below whatever directory structure is configured in the file `iah.def.php`

All examples in this manual assumes IAH under Apache environment.

Default structure of IAH in ABCD

ABCD package comes with IAH preinstalled and configured. If you want to install IAH as an independent application, please refer to the IAH page at VHL Model Site (<http://bvsmodelo.bvsalud.org/php/level.php?lang=en&component=31&item=3>)

IAH application distributed in ABCD package contains all the files and directories necessary for the interface, but with a small difference between the version for Windows and the version for Linux.

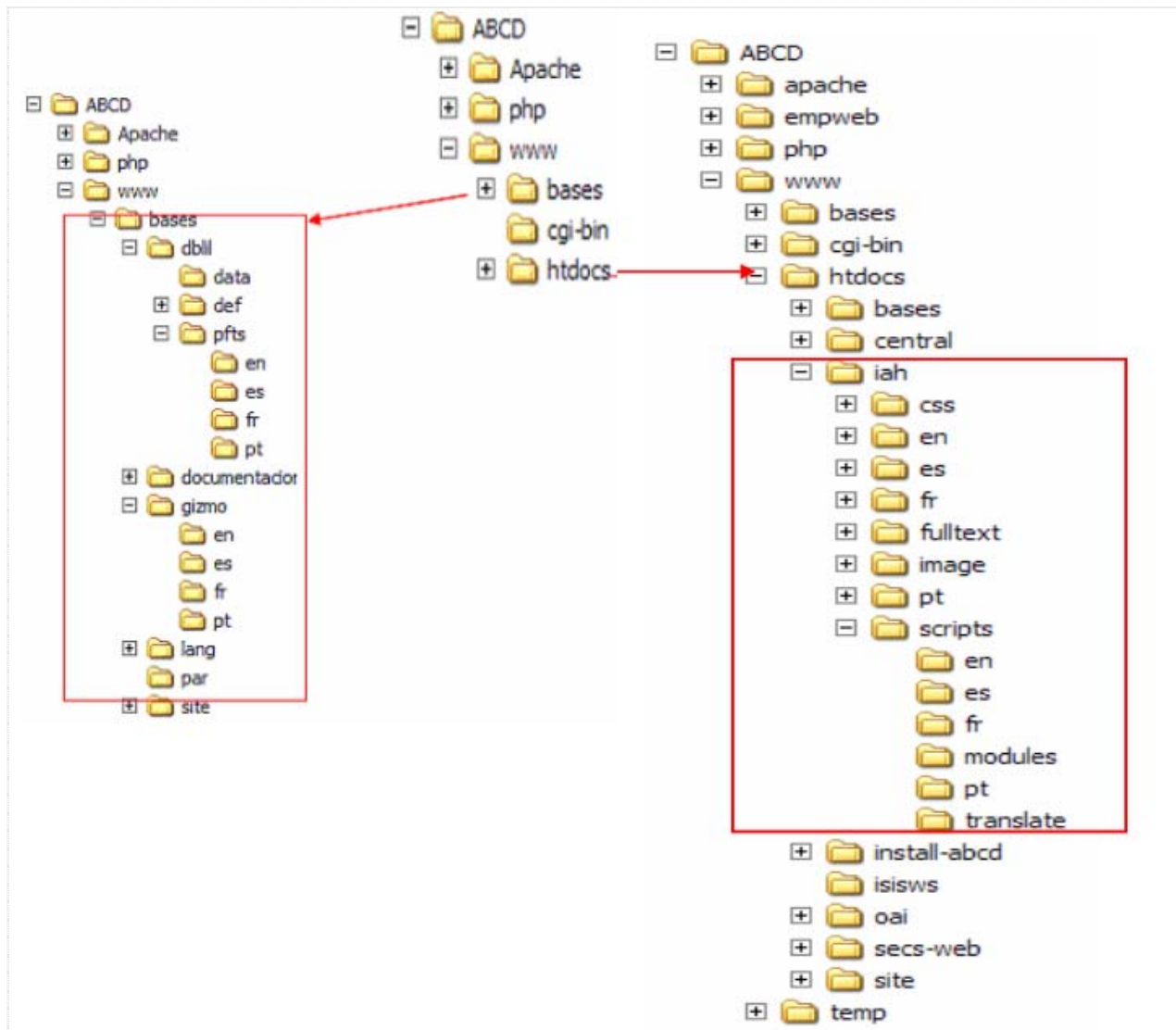


Figure 1: Organization of directories for the Windows and Linux distributions

Files and directories of the IAH package

BASES/

Below this directory there are two general subdirectories: `gizmos/`, and `par/` and a subdirectory for each of the databases `<dbn>` informed to IAH; for example the demo database `DBILIL`. Other directories below BASES correspond to ABCD application, but they are not part of IAH application.

- In `gizmos/` there are various ISIS databases (`.mst`, `xrf` and `id` files) which complement the interface, the conversion table for characters, `gizmo` and related tasks.
- In `par/` there are various files with extension `.def` which gives to IAH all the needed parameters for searching and displaying the databases. Each database requires its own `.def` file.

Below each `<dbn>` directory there are four subdirectories belonging to IAH:

- In `data/` are the master and inverted file (data and dictionary) of the database, plus some auxiliary files [optional]
- In `fst/` are found the field selection table for the database and the stopword file [optional].
- In `isos/` directory is the source data to update/replace the actual database
- Finally in `pfts/` are the various display formats for the database.

HTDOCS/IAH

This directory contains eight subdirectories (`css/`, `en/`, `es/`, `fr/`, `image/`, `pt/`, `scripts/` and `fulltext/`), and two configuration files. These two-letter directories (corresponding to the ISO code for languages) each one contains the language-specific messages, banner, icons, etc., of IAH. The directory `fulltext` is only for demo, you can delete it after the system is working.

In the directory `htdocs/iah/scripts/` there are six subdirectories (`en/`, `es/`, `fr/`, `modules/`, `pt/` and `translate/`), a configuration file for IAH (`iah.def.php`) and the IsisScript code of the application (files with extension `.xis`). In the subdirectories of each language are the format files which form the pages of the interface in that language.

The subdirectory `htdocs/iah/scripts/modules` contains optional modules and language extensions that permit the additional use of new functions with more flexibility, as well as customisations.

The subdirectory `translate/` contains a translation tool aimed to help users in translating pages to other languages using batch files, the MX utility and some gizmo files (master files).

In the directory `htdocs/iah/`, each of the subdirectories `en/`, `es/`, `fr/`, and `pt/` contains two subdirectories: `image/` and `help/`. In the first one are the image files used by the interface such as icons, buttons, banners, logos etc. In the second one are the files containing the help texts of the interface in the respective languages in HTML format.

The subdirectory `htdocs/iah/css/` contains the stylesheets of IAH interface.

The subdirectory `htdocs/iah/image/` contains the image files language independent used by the interface.

The file `htdocs/iah/index.php` contains the initial page of the IAH interface after its installation and configuration and allows access to the databases.

In the subdirectory `htdocs/iah/fulltext/` there are files for providing access to full texts for the demo database.

CGI-BIN

In this directory is found the executable file of WWWISIS (`wxis.exe`) for interpreting and executing the application IAH written in IsisScript.

There is also the full package of CISIS utilities for maintenance of databases using scripts (`.bat` or `.sh`) or from the command line. It is highly recommended that the system has set in the environment a path to this directory in order to have available access to this files from every place in the server.

Understanding the configuration of IAH

Contents of the file iah.def.php

IAH.DEF.PHP is a text file composed of four sections, whose functions are detailed in table 2.

TITLE OF THE SECTION	DESCRIPTION
[PATH]	Indicates the location of the directories and files of the interface
[APPEARANCE]	Defines settings relating to the appearance of the interface
[HEADER]	Defines images and links for the head of the pages
[IAH]	Indicates the e-mail address of the person responsible and some configuration options of the interface

Table 2: Sections and functions of iah.def.php

Example of iah.def.php default of ABCD package

```
<?php

[PATH]

PATH_DATA=/iah/
PATH_CGI-BIN=/ABCD/www/htdocs/iah/scripts/
PATH_DATABASE=/ABCD/www/bases/
PATH_DEF=/ABCD/www/bases/par/

[APPEARANCE]

/* Please adjust /css/styleSheet.css */

[HEADER]

LOGO IMAGE=logoabcd.jpg
LOGO
URL=^1http://www.bvs.br/^2http://www.bvsalud.org/^3http://www.virtualhealthlibra
ry.org/^4http://www.virtualhealthlibrary.org
HEADER IMAGE=online.gif
HEADER
URL=^1/iah/pt/index.htm^2/iah/es/index.htm^3/iah/en/index.htm^4/iah/fr/index.htm

MANAGER E-MAIL=ia@my-server.domain
REVERSE MODE=ON
MULTI-LANGUAGE=ON
AVAILABLE LANGUAGES=pt, es, en, fr
?>
```

PATH section

The definitions contained in this section are:

- **PATH_DATA=**
Indicates the path relative to the root of the application in the webserver in which the static pages are located (**IAH**)
- **PATH_CGI-BIN=**
Indicates the absolute path of the directory for IsisScripts

(e.g. /ABCD/www/htdocs/iah/scripts/)

- **PATH_DATABASE=**
Indicates the absolute path of the directory for databases on the server
(e.g. /ABCD/www/bases/iah)
- **PATH_DEF=**
Indicates the absolute path of the directories that contains the database configuration file
(e.g. /ABCD/www/bases/par/)

Note: all paths should include initial and final slash.

APPEARANCE section

The definitions of colours, styles, fonts, etc. of IAH interface, are defined in four files in /css/stylesheet.css

```
/css/  
  general.css  
  layout.css  
  styles.css  
  stylesheet.css
```

The "css" directory contains the IAH Site stylesheets. To customize the interface, the files of the this directory should be changed. This directory hosts four files CSS that will be detailed below in this section.

Style sheet structure

IAH style sheets are divided into 4 different files in the `htdocs/iah/css/` directory:

- **general.css** – this CSS has the general site attributes, such as color and standard font type used in the site.
- **layout.css** – in layout we have definitions of sizes, element positions, margins, etc.
- **styles.css** – in this CSS there are definitions of IAH Site style, element colors, banner images, typographic styles, borders, etc.
- **stylesheet.css** – is the main CSS which include the other style sheets files.

The style sheets are loaded by the system, in the previously mentioned order. Thus, any alteration to the last CSS, i.e. `stylesheet.css` will overwrite the element defined in previous CSS's.

HEADER section

The definitions which make up this section are the following:

- **LOGO IMAGE=**
Determines the image file of the logo which is used in the head of the HTML pages. In general this type of file is located in the subdirectory image below `htdocs/iah/<lang>/`, where `<lang>` equals to 'pt', 'es' and 'en'.
- **LOGO URL=**
Determines the URLs of the websites which host the IAH interface in the three available languages

- **HEADER IMAGE=**
Determines the image file which heads the HTML pages. In general this type of file is located in the subdirectory image below htdocs/iah/<lang>/, where <lang> equals to 'pt', 'es' and 'en'.
- **HEADER URL=**
Indicates the URLs of the initial pages of IAH (the page giving the selection of databases) in the three available languages

IAH section

The definitions which make up this section are the following:

- **MANAGER E-MAIL=**
Determines the e-mail address of the person responsible for the maintenance of IAH. Users are informed of this address in case of errors in the system
- **REVERSE MODE=**
Determines the order of presentation of the results of searches. The possible values are: in ascending order (*OFF*), or descending order (*ON*) of MFNs
- **MULTI-LANGUAGE=**
Indicates whether the user can or cannot change the language of the interface. The possible values are: *OFF* for change is prohibited, and *ON* for change is permitted, between the languages English, French, Spanish and Portuguese.
- **AVAILABLE LANGUAGES=**
Indicates the assigned codes of the available languages, separated by coma. The order of the languages corresponds to the subfield numbers used for URL, and FORMATS

How to configure the file iah.def.php

If you need to configure IAH with different settings, refer to the official manual (<http://bvsmodelo.bvsalud.org/php/level.php?lang=en&component=31&item=3>)

Initial testing of the interface

Using an Internet browser (Internet Explorer, Mozilla, Netscape etc.) type in the the address box the URL in which IAH was installed, e.g.

```
http://my-server/iah/index.php
```

where *my-server* is the name of the domain where the package is installed.

It is possible to access an installation in local mode using **localhost** or **127.0.0.1** as the name of the domain, like:

```
http://localhost:9090/iah/index.php
```

From this page it is possible to select the example database **DBLIL** which accompanies ABCD package. **DBLIL** is a sample of LILACS database of BIREME.

It is suggested that you perform a search in the example database **DBLIL** in order to confirm that the interface is functioning correctly before proceeding with the process of installation and configuring your own databases, personalizing the layout, etc.

In case of error you can use the URL below:

```
http://my-server/cgi-bin/wxis.exe?hello
```

`http://my-server/cgi-bin/wxis.exe?IsisScript=getenv.xis`

Your own databases

To install and configure your own database, you need to follow the following steps:

How to install a new database

1. Transfer to the directory `bases/wrk` the file(s) in ISO format of the desired database(s)
2. Transfer to the directory `bases/<your_base>/data` the FST file(s) of the database(s)
3. Transfer to the directory `PFTS (bases/<your_base>/pfts/?/)` the display format files of the chosen database(s). The "?" corresponds to the language subdirectories of the interface (`en/`, `es/`, `fr/`, `pt/`).

Notes: You have to provide a full set of formats for each available language in IAH.

Commands for indenting and vertical and horizontal space (characteristics of the CDS/ISIS format language) must be changed to their equivalents in HTML. Fields which contain symbols for control of filing (e.g. `<A>`, `<The>`, `<La>`, etc.) must have as their display mode *heading* (mhl).

4. Execute the database configuration file in the command line of DOS or Linux:
 - a) Change to the database directory (e.g. `/home/bases/`)
 - b) Execute the command `setupdb` (shell or batch file – depending on the operating system), giving the name of the ISO file (including the extension), name of the FST file FST (including the extension) and the name of the resultant database (e.g. `setupdb.bat base.iso base.fst base`).

Notes: If the database has been set up with the ANSI (Windows) character table, use the fourth parameter, `ANSI`, in the instruction `setupdb`. The default for inversion is ASCII.

A generic `setupdb.bat` or `setupdb.sh` will be added in a later release

Create a copy of the file `DBLIL.def` (which is in `/bases/par/` directory) with the name of your database, e.g. `video.def` and edit it for the requirements of your database, in accordance with the following section "*How to configure a new database*".

5. Add to the page for access to databases a link to this new database conforming to one of the following models, depending on your Site:

You use an independent page to invoke the database

```
<p> <A HREF = "http://<my_server>/cgi-  
bin/wxis.exe/iah/scripts/?IsisScript=iah.xis&base=<YOUR_BASE>&  
lang=en"> Database</A> - Text describing the database</p>
```

If you use ABCD Site as content integrator, use the link

```
http://%HOST%/iah/?base=DBLIL&lang=<lang>
```

The terms in bold should be altered to reflect the details of your own database. The parameter `base=YOUR_BASE` should be uppercase.

How to configure a new database

To configure a database in IAH it is necessary to alter some parameters in the database definition file (`BASE.def`), whose name is defined by the user during the installation (7.1, item 5) and which can be based on the definition of the LILACS database (`DBLIL.def`, or any other you prefer best).

Note: Linux operative system is case sensitive, therefore take care of the capitalization of the name of the file `BASE.def`.

`BASE.def` is a text file composed of six sections, whose functions are detailed in table 3.

TITLE OF THE SECTION	DESCRIPTION
[FILE_LOCATION]	Location of the database files, display formats and of export of data
[INDEX_DEFINITION]	List of definitions of indexes that can be accessed in searching
[APPLY_GIZMO]	List of files used for the global change of contents
[FORMAT_NAME]	List of definitions of display formats available
[HELP_FORM]	List of help files for the search forms
[PREFERENCES]	List of search forms available

Table 3: Sections and functions of the file `BASE.def`.

Section FILE_LOCATION

In this section you should define the logical names of databases, inverted files, files for display formats and conversion of data, all with their respective physical paths.

For each database, you need to define, at least, three kind of files: (1) the database, (2) the inverted file(s), and (3) the display format files. For the definition of the database file the logical name **DATABASE** must be used.

Each definition is preceded by the instruction **FILE** in the general form:

```
FILE LOGICAL_NAME.*=/directory-1/directory-n/file_name.*
```

Using the definition of a database with the name "video" as example, the declaration results in:

```
FILE DATABASE.*=/ABCD/www/bases/video/data/video.*
```

Note: If you work in a Linux environment the names of your files in the declaration should match the case of the physical files. Do not use diacritical signs to name a database, for example use `video` and not `vídeo`.

To ensure that the interface is portable to any directory structure, three variables have been predefined:

1. `%path_database%`
2. `%path_cgi-bin%`
3. `%lang%`

The variables `%path_database%` and `%path_cgi-bin%` get their values from the settings of `PATH_DATABASE` and `PATH_CGI-BIN`, respectively, of the file `iah.def.php` while the variable `%lang%` receives the identifying letters of the language (`en`, `es`, `fr`, `pt`) according to the selection made by the user through the interface.

Hence the definition of the database file can be written like this:

```
FILE DATABASE.*=%path_database%video/data/video.*
```

In the same way, the definition of a display format can be written:

```
FILE standard.pft=%path_database%video/pfts/%lang%/lildhtm.pft
```

Note that the standard structure of the ABCD package is used in all the examples.

Section INDEX_DEFINITION

In this section you should define the indexes available for the search. It is possible to mount an indexing FST or an FST with prefixes. The use of a fixed descriptive element of index excludes or includes other variables, depending on the group to which they pertain. The elements referring to the language are neuter and optional.

Each definition is preceded by the term INDEX, followed by a prefix which can have various descriptive elements identified in subfields following table 4.

Element	Group	Description
^1 ... ^9	standard	Name of the index in the n th language declared in other section (1)
^d	standard	Marks the default index (2)
^f	standard	Indicates availability of the field only in the advanced form (content: "A") (3)
^t	standard	Type of the index ("short" for short index, "hidden" for hidden index) (3)
^g	standard	Defines the gizmo for executing in the keys of the inverted file (3)
^x	prefixed	Identifies the prefix which will be used in search strategies
^y	prefixed	Identifies the logical name of the inverted file
^u	prefixed	Identifies the prefix to use
^m	prefixed	Identifies the prefix referring to the form or note associated with the search index. It works in conjunction with the definition of specific notes for the fields in the section HELP_FORM

Table 4: Components of the definition of indexes.

(1) Optional element in accordance with the selected language.

(2) Mandatory element for the default index. Only one index should be default.

(3) Optional element.

As a general form, we have:

```
INDEX ??=^1<Name>... ^9<Name>[^d*]^x?? ^u??^yDATABASE^m??^fA^t<s|h>^g<file>
```

where the ?? sign stands for two letter prefix used.

So, the definition of a word index to present as default could be:

```
INDEX TW=^1Palavras^2Palabras^3Words^d* ... etc.
```

In the example above, the index shown as "Words" in English, would be the default and would perform a search in all the fields defined in the FST. This definition also implies an FST without field prefixes.

For an index of subject descriptors identified by the prefix "MH" in the FST and available only in the English language, the declaration would be:

```
INDEX MH=^3Subject descriptor^xMH ^uMH_^yDATABASE^mMH_
```

Note: The blank space after the ^xMH and before the ^u is mandatory

For an index of words from the abstract identified by the prefix "AB" in the FST and with reference to an inverted file with the name "ABSTRACT" we would have in the files section of the definition:

```
FILE ABSTRACT.*=%path_database%abstract/def/abstract.*
```

and in the definition of indexes:

```
INDEX AB=^1Palavras do resumo^2Palabras del resumen^3Abstract  
words^xAB ^uAB_^yABSTRACTmAB_
```

Section APPLY_GIZMO

In this section you should indicate the files for the global changes of strings of characters contained in the databases (GIZMO files), permitting conversion of character tables, coding/decoding of data, compression of information, etc.

The logical names must be defined in the section FILE_LOCATION

Each definition should be preceded by the term GIZMO, with the general form:

```
GIZMO LOGICAL_NAME
```

The standard IAH package contains some predefined GIZMO databases:

LOGICAL NAME	Description
GANS850	Performs the conversion of ANSI characters to their equivalences in ASCII 850.
G850ANS	Performs the conversion of ASCII 850 characters to their equivalences in ANSI.
QLFANS	Performs the decoding of the codes used in DeCS descriptors in accordance with the ANSI (Windows) table. Available in three languages.
QLF850	Performs the decoding of the codes used in DeCS descriptors in accordance with the ASCII 850 (DOS) table. Available in three languages.
LANGS	Performs the decoding of the language codes in accordance with the ANSI (Windows) table. Available in three languages.
GIZMOTL	Performs the decoding of the type of literature codes in accordance with the ANSI (Windows) table. Available in three languages.
GIZMONB	Performs the decoding of the bibliographic level codes in accordance with the ANSI (Windows) table. Available in three languages.

Table 5: Table of standard GIZMOS in the IAH package.

Note that each gizmo table matches the earlier definition in the file DBLIL.def of the example included with the package:

```
FILE GIZMOTL.*=%path_database%gizmo/%lang%/gizmotl.*  
FILE GIZMONB.*=%path_database%gizmo/%lang%/gizmonb.*  
FILE LANGS.*=%path_database%gizmo/%lang%/lang.*  
FILE G850ANS.*=%path_database%gizmo/g850ans.*  
FILE GANS850.*=%path_database%gizmo/gans850.*
```

```
FILE QLFANS.*=%path_database%gizmo/%lang%/qlfans.*
FILE QLF850.*=%path_database%gizmo/%lang%/qlf850.*
```

You can specify the database GIZMO ASC2ANS for converting data recorded in the database in ASCII format into ANSI as follows:

```
GIZMO=ASC2ANS
```

Note: There are gizmos declared in section FILES and gizmos declared in section APPLY_GIZMOS. The difference is that gizmos in FILES are used by default when reading the database, but gizmos declared in APPLY_GIZMO section are used when displaying fields in browsing.

Section FORMAT_NAME

In this section you define the logical names of the available display formats and their descriptions in the available languages of the instantiation of IAH

Each definition is preceded by the term **FORMAT**, followed by the real name and of the descriptions for each language (^1, ^2, ... ^9, corresponding to the assigned languages in iah.def.php) like this:

```
FORMAT xxxxx=^1nonononon^2nonononon^3nonononon
```

Note: ^0 (zero) is not enabled to assign languages

The installation package comes accompanied by four display format files for the language of the interface for the DBLIL demo, complying with LILACS methodology.

The four formats for LILACS are: long, detailed, citation and title. The default format is citation. The files (with the extension .pft) defined for each of the formats must be declared in the section [FILE_LOCATION].

The definition of the detailed format is declared like this:

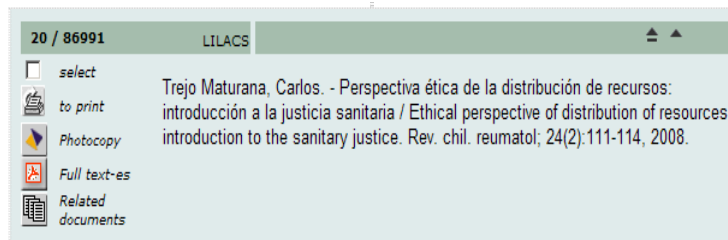
```
FORMAT detailed.pft=^1Detalhado^2Detallado^3Detailed^4Détailée
```

The declaration FORMAT DEFAULT specifies which is the default for the interface and should preferably be the last declaration of this section. Note that the format must have been declared previously:

```
FORMAT DEFAULT=detailed.pft
```

FILE SHORTCUT.IAH=

This feature enables the database administrator to add optional functions and services in the left column of the display, that are prepared using display formats commands.



Section HELP_FORM

In this section you define the HTML files for help and the explanatory notes for each type of form (free, basic and advanced) and/or the indexes available in the interface. It is assumed

that these files are located under the root of IAH in the subdirectory help below each language (en, es, fr, pt).

The content of these files can be generic, serving all the elements of a particular category (form or index), or specific, for determining one element within the category. For example, you can define a help file which serves all the available indexes and additionally a help file that gives more detail of one of them.

To define a generic help text, the declaration should be preceded by the term **HELP FORM** or **HELP INDEX** and followed by the physical name of the file.

For example:

```
HELP FORM=help_forms.html
```

```
HELP INDEX=help_indexes.html
```

To define a specific help text, the declaration should include the type of search form (F: free, B: basic and A: advanced) and/or index (in accordance with the section INDEX_DEFINITION).

For example:

```
HELP FORM F=help_form_free.html
```

```
HELP INDEX TW=help_index_words.html
```

To define a text for *explanatory notes*, the declaration should be preceded by the term **NOTE FORM** or **NOTE INDEX** in place of HELP FORM and HELP INDEX. The rest of the procedure is identical.

```
NOTE FORM F=note_form1_lilacs.htm
```

```
NOTE INDEX TW=note_index_words.html
```

Section PREFERENCES

In this section you define some options of the interface for the particular database. The declarations contained in this section are the following:

- **AVAILABLE FORMS**

Determines the set of forms availables, the first one listed being the default in the initialization. The forms are identified by letters as shown in Table 6.

Identifier	Form
F	Free
B	Basic
A	Advanced

Table 6: Identifiers of Forms.

The definition is preceded by the term **AVAILABLE FORMS**, followed by the list of identifiers of forms separated by commas, as:

```
AVAILABLE FORMS=F,A
```

This declaration informs IAH that there are two search forms available (Free and Advanced), with the free form being the default in the initialization.

Free form presents a simple search area (Google like), no explicit boolean operator are enabled.

Basic and Advanced presents a guided search interface, with some boolean capabilities. The difference between B and A is the number of indexes availables.

- SEND RESULT BY EMAIL

Determines whether or not the sending of search results by email is enabled. The possible values are ON and OFF. For this option to function correctly the user should install and configure an SMTP program and create a routine for receiving the parameters sent by the interface.

```
SEND RESULT BY EMAIL=OFF
```

The command which is executed when this option is activated should be given in the file *sendmail.conf* of the directory root of the IAH (/home/user/htdocs/iah/).

Under Windows the content of the file *sendmail.conf* could be:

```
EXECUTE=blat <file> -subject Results -to <mailto> -q -html
```

Note: It is possible to insert many other parameters. The data between < ... > is provided by IAH.

Example in Windows using Blat (it is assumed the program is active)

```
blat <file> -subject Results -to <mailto> -sender mylibrary@something.com  
-from mylibrary@something.com -embed some.gif -embed some.jpg  
-html -q -log log.txt
```

Under Linux the content would be:

```
EXECUTE=/home/user/iah/cgi-bin/email.sh <mailto> <file>
```

Example of email.sh

```
export TEMP_FILE=/var/tmp/$$.@@@  
echo "From: test@test.com" > $TEMP_FILE  
echo "To: $1" >> $TEMP_FILE  
echo "Subject: Results " >> $TEMP_FILE  
echo "Mime-Version: 1.0" >> $TEMP_FILE  
echo "Content-type:text/html" >> $TEMP_FILE  
cat $2 >> $TEMP_FILE  
/usr/lib/sendmail -t -oi < $TEMP_FILE  
rm $TEMP_FILE
```

Notes: The program responsible for sending e-mail does not form part of the IAH package.

Under Linux, the shell shown in the example is responsible for passing the two required parameters to the chosen mail program.

- NAVIGATION BAR

Determines whether the navigation bar between the retrieved documents should be shown or not. The possible values are *ON* and *OFF*.

```
NAVIGATION BAR=ON
```

- DOCUMENTS PER PAGE

Determines the maximum number of documents shown in the results page of a search.

```
DOCUMENTS PER PAGE=20
```


- FEATURES

Indicates additional characteristics of IAH availables for use. Currently, the package only supports the XML standard in the export of records.

For example:

```
FEATURES=XML
```

Note: This feature requires the existence of a file like `lilCitationXML.pft` include in the demo package and defined in a line in `FILE` section of `BASE.def`

Management of multiple databases with a different layout

A single instance of IAH can hold a large quantity of databases and yet permit the layout, the colours and the output format of data be changed.

When you want to add new databases in IAH (e.g. rare books and institutional – referred to here as "rare" and "videos") with clearly distinct visual identities, you should start with the preparation of the environment in which you replicate the parameters of the installation model.

1. Create a subdirectory for each database in the area of database, e.g.

```
bases/rare
```

```
bases/videos
```

2. Copy to the `bases/par/` directory the configuration file `dblil.def` to each database (`/bases/par/raros.def` and `/bases/par/videos.def`);
3. Copy the subdirectories `data/`, `def/` and `pfts/` below `/base/<database>` for each subdirectory (`bases/rare` and `bases/videos`)
4. Copy the subdirectories `en/`, `es/` and `pt/` abajo of `/bases/<database>/pfts` for each subdirectory of the area of databases (`bases/rare/pfts` and `bases/videos/pfts`)

Note: It is not required to copy the subdirectories of the languages that are not in use.

With the structure prepared, you can commence the customization of each new virtual instance of IAH, as follows:

Note: In the steps below it is shown how to configure the instance "rare" which was previously prepared.

1. Edit the file `bases/par/rare.def`, changing the lines below, described in the section `FILE_LOCATION`:

from:

```
FILE DATABASE.*=%path_database%dblil/data/dblil.*
```

```
FILE DATABASE.XML=%path_database%dblil/pfts/lilXML.pft
```

```
FILE standard.pft=%path_database%dblil/pfts/%lang%/lillhtm.pft
```

```
FILE detailed.pft=%path_database%dblil/pfts/%lang%/lildhtm.pft
```

```
FILE citation.pft=%path_database%dblil/pfts/%lang%/lilchtm.pft
```

```
FILE mes.pft=%path_database%dblil/pfts/%lang%/mes.pft
```

```
FILE citation.xml=%path_database%dblil/pfts/lilCitationXML.pft
```

```
FILE title.pft=%path_database%dblil/pfts/%lang%/lilthtm.pft
```

```
FILE SHORTCUT.IAH=%path_database%dblil/pfts/%lang%/shortcut.pft
```

```
FILE descriptores.pft=%path_database%dblil/pfts/%lang%/descriptores.pft
```

```
FILE GIZMOTL.*=%path_database%gizmo/%lang%/gizmotl.*
```

```

FILE GIZMONB.*=%path_database%gizmo/%lang%/gizmonb.*
FILE LANGS.*=%path_database%gizmo/%lang%/lang.*
FILE G850ANS.*=%path_database%gizmo/g850ans.*
FILE GANS850.*=%path_database%gizmo/gans850.*
FILE QLFANS.*=%path_database%gizmo/%lang%/qlfansi.*
FILE QLF850.*=%path_database%gizmo/%lang%/qlf850.*

```

to:

```

FILE DATABASE.*=%path_database%rare/data/rare.*

FILE standard.pft=%path_database%rare/pfts/%lang%/lillhtm.pft
FILE detailed.pft=%path_database%rare/pfts/%lang%/lildhtm.pft
FILE citation.pft=%path_database%rare/pfts/%lang%/lilchtm.pft
FILE mes.pft=%path_database%rare/pfts/%lang%/mes.pft
FILE title.pft=%path_database%rare/pfts/%lang%/lilthtm.pft
FILE SHORTCUT.IAH=%path_database%rare/pfts/%lang%/shortcut.pft
FILE descriptores.pft=%path_database%rare/pfts/%lang%/descriptores.pft

```

2. Copy the database "rare" (*rare.mst* and *rare.xrf*) to the directory defined in the option FILE DATABASE of the file bases/rare/data/rare.def
3. Copy the FST bases/dblil/data/dblil.fst to bases/rare/data/rare.fst
4. Generate the inverted file for the database with rare.fst using the line below as a model:
 - a) for Windows:

```

mx rare actab=ansiac.tab uctab=ansiuc.tab fst=@fsts\rare.fst
fullinv=rare now -all

```
 - b) for Linux

```

./mx rare actab=ansiac.tab uctab=ansiuc.tab fst=@fsts/rare.fst
fullinv=rare now -all

```

Now, you need only to construct the URL for the database with the parameters changed to access the new virtual instance RARE.

Let us consider, therefore, the basic URL of IAH for the example database DBLIL in the English language that comes with the IAH package:

```
http://domain/cgi-bin/wxis.exe/iah/scripts/?IsisScript=iah.xis&lang=en&base=dblil
```

We can separate the terms of the URL into blocks like this:

Term	Code
protocol	http:
domain	//domain/
program	cgi-bin/wxis.exe
root of the application	/iah/scripts/
parameter 1: application	IsisScript= iah.xis
parameter 2: language	lang=en
parameter 3: database	base=dblil

To construct the appropriate URL for the database "RARE", you should follow these steps:

1. Change the name of the database in parameter 3:

```
base=rare
```

The table with the new parameters will be:

Term	Code
protocol	http:
domain	//domain/
program	cgi-bin/wxis.exe
root of the application	/iah/scripts /
parameter 1: application	IsisScript=iah.xis
parameter 2: language	lang=en
parameter 3: database	base= rare

2. Reconstructing the URL, this becomes:

```
http://domain/cgi-bin/wxis.exe/iah/scripts/?IsisScript=iah.xis&lang=en&base=rare
```

Now that there is a separate area configured for the new database (RARE), the image files (`htdocs/iah/??/image`) and the support and help texts (`htdocs/iah/??/help`) can be changed without affecting changes to the original files or other virtual instances.

It is also possible to change the definitions of colour and the names of logos defined for the interface, in the case the user wants to differentiate each instance, through the parameters defined in the sections APPEARANCE and HEADER.

How to use the IAH interface

Any search in the databases is performed through a form in which the user enters his/her "search expression", made up of words, terms and boolean (or logical) operators.

After the required database has been selected, the IAH interface presents the "free" form (adopted as the default of the installation), with other forms available: basic and advanced.

The examples and figures are based on IAH configured for the database DBILIL of ABCD.

Free form

This searches words in all the fields that are inverted word by word.

Important remarks

- When there is more than one word you should separate them by a single space, selecting with the radio button one of the available operators: AND (to relate) or OR (to sum). Do not include this words in the search expression.
- If you want to truncate terms or words partially, use the symbol (\$) at the end of the root, for example: `infect$` retrieves `infection`, `infections`, `infectious` etc.
- It is important to consider the synonyms and variations of the search words (e.g. `cerebral`, `brain`).

Example 1: Search in the DBILIL database about health and environment and sustainable development

Consider only the most significant words for your search and type them into the available space. In this example you should leave the operator AND selected so that IAH searches for articles that contain all the words entered.

Figure 2: Example of completing the field for a free form.

Basic and Advanced Forms

The **Basic Form** permits searching in the principal search fields of the databases, amongst them: words, descriptors, limiting by subject, author, journal and language, so offering access to the indexes of those fields which help to assemble the search expression.

The **Advanced Form** permits searching in all the indexed fields of the datadatabase and has the same resources as the basic form.

To go to the basic or advanced forms, you have to click the name of the **form** in the upper right part of the free form.

Figure 3: Location of the options for of type of search form.

How to search

1. **Do not type anything** in the white boxes.
2. Select one of the search fields, click the button which matches in the box showing the fields, and click the index button on the same line.

Figure 4: Selection of search field.

Notes: When you have more than one word or term in the same line, you must use one of the boolean operators (AND, OR, AND NOT) between each term.

The operator **AND** is defined previously as default between one line and another, but can be altered by the user according to the requirements of the search.

The screenshot shows a search form with three lines of input. Line 1 contains 'public\$'. Line 2 contains 'and' in a dropdown menu followed by 'america\$'. Line 3 contains 'and' in a dropdown menu followed by 'EN'. To the right, there are three dropdown menus for 'in field': 'Subject descriptor', 'Words', and 'Language'. Below the input fields are three buttons: 'config', 'limpar', and 'Search'.

Figure 5: Example of a search expression entered in the basic form.

In this example using the **AND** operator, the system relates the two subjects and retrieves only articles in English.

Display of the results

Once the search has been executed, the first part of the page shows information about the database searched, the search expression, and some of the references retrieved in the search in groups of 20 records, in the display format indicated in the **CONFIG**. (In the example below the display is in the long format).

The screenshot shows a search result page. At the top, there are navigation buttons: 'your selection', 'send result', 'new search', 'config', and 'page bottom'. Below these, the search details are displayed: Database: DBLIL, Search: public\$ and america\$ and EN, References found: 19 [Refine the search], Showing: 1..19 in format [Detailed]. The page is labeled 'page 1 of 1'. The main content area shows '1 / 19' and 'DBLIL'. On the left, there are icons for 'select', 'print', and 'Full text'. The record details include: Id: 170, Author: Anon, Title: Regional consultation for the Americas on health research for attainment of the Millennium Development Goals (MDGs) ...-, Source: s.l; s.n; s.d. 6 p. ., Conference: Present in: Regional consultation for the Americas on health research for attainment of the Millennium Development Goals (MDGs), México, DF, May 24-26 2004., Descriptors: Pesquisa/tend, Saúde Pública/tend, - Américas, Países Desenvolvidos, Países em Desenvolvimento, Acesso à Informação, Limits: Humanos, Electronic Internet, Medium: .

Figure 6: Example of a search result.

Note: The number of records per page can be altered through the declaration DOCUMENTS PER PAGE in the definition file of the database.

The principal commands and resources available in this page are the following:

[[Refine the search](#)] – changes the search form to the previous search expression.

1. [new search](#) - changes the search form to a new search.

2. **your selection** - lists the references (or records) previously selected.
3. The selection is achieved by clicking in the area [selection] available in the left corner of each reference displayed, as shown in the picture below where the reference number 1 is selected. To cancel the selection, it is sufficient to click again within the area.
4. **send result** -permits printing or capturing the references or records retrieved in the search, selected or not, for sending the references by electronic mail (e-mail).
This option can include:
 - a) all the references retrieved in the search
 - b) a sequential range of references retrieved, e.g. 1 to 50
 - c) The references selected previously in the results page of the search
5. In the option to send for printing, the system generates a list with the option defined by the user. From this list you should use the print or capture command of your web browser.
 - To print – use the option Print of your web browser
 - To customize the print outs of IAH with different logo and footer of the institution, should modify the following files: ahlist.pft, afoot.pft in the folder htdocs/iah/scripts/%lang%/
 - To capture in text format – use the option File, Save As of your web browser
6. **config** - command that permits changing the format of display/listing of the references retrieved in the search; changing the language of the interface; and the option to include or not the navigation bar.

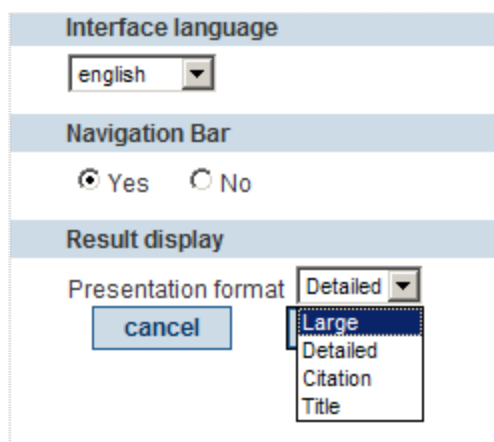


Figure 7: Window for configuring the interface.

The formats available are the following:

- **long**: includes bibliographic data of the reference and abstract
- **detailed**: includes bibliographic data of the reference, summary and descriptors (or keywords)
- **title**: includes only the title of the article/document
- **citation**: is the format of bibliographic reference, without abstract (default format).

7. Select the desired format and confirm (or cancel) with the button **cancel** **apply**
8. To navigate between the pages, use the pagination bar of the results.

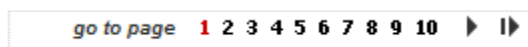


Figure 8: Pagination bar of the results.

Each page of results includes a maximum of 20 references: it is possible to configure this value during and after the installation. To navigate amongst the reference, or within the same page, use the array of arrows available in the navigation bar of the records



Logical operators in the search

The logical operators or boolean operators are used to relate terms or words in a search expression. They combine two or more terms in one or more search fields. The boolean operators accepted in IAH are:

- AND, OR, AND NOT, parenthesis, \$ (truncation sign).
- IAH accepts either the reserved words or the corresponding symbols: * + ^
- Proximity operators of CDS/ISIS are not implemented in IAH.

You can use boolean operators in Advanced Forms in IAH.

Search fields

Words

The **words** search field is considered a free-text field, formed of single words. Therefore, you should not include in your search formulation compound terms and phrases if the field was not indexed by words. Note: you should or should not use articles, prepositions, depending on the inclusion or not of a stopword file (.stw) associated with the database.

Example – search for “*complications of hepatic cirrhosis*” in the DBILIL database

Enter one or more words

public\$ america salud

All words (AND) Any word (OR)

config Search


Figure 8: Example of a free search with operator AND (DBILIL database).

Descriptor

The **descriptor** field is the preferred route for an efficient search with the best results. This field contains terms which represent the subject of the article or document, known as descriptors, keywords, uniterms, subject headings, MeSH terms (Medical Subject Headings), or DeCS terms. [“Keyword” is somewhat ambiguous in English as it is also sometimes used to mean significant words in the title, abstract, etc.]

Example: search for the subject “SINDROME DE IMUNODEFICIENCIA ADQUIRIDA ”

Step 1- in the basic or advanced forms, **do not type in the white boxes.**

1. Select in line 1 of the **descriptor** search field, having clicked the button which matches in the box reserved for the fields
2. Click the image  index to open the page for access to the index of the field selected in the form.

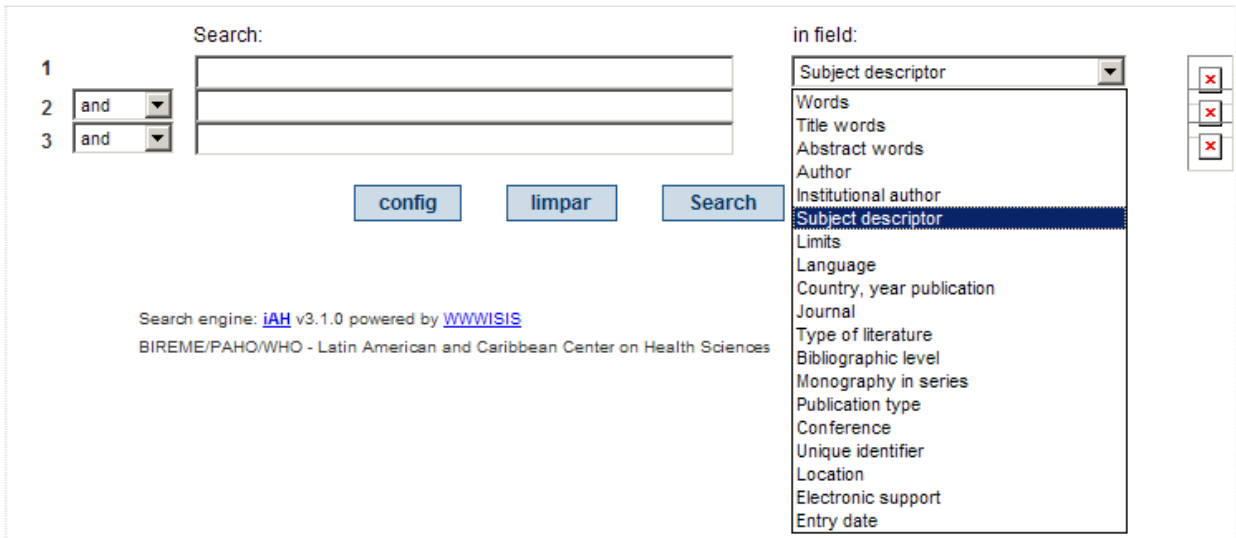


Figure 9: Selection of the descriptor in the box reserved for the search fields and the button for access to the index

Step 2 – in the page for access to the index type in the appropriate space the first word or element of your search (in the example **syndrome**), and click on **show index**.

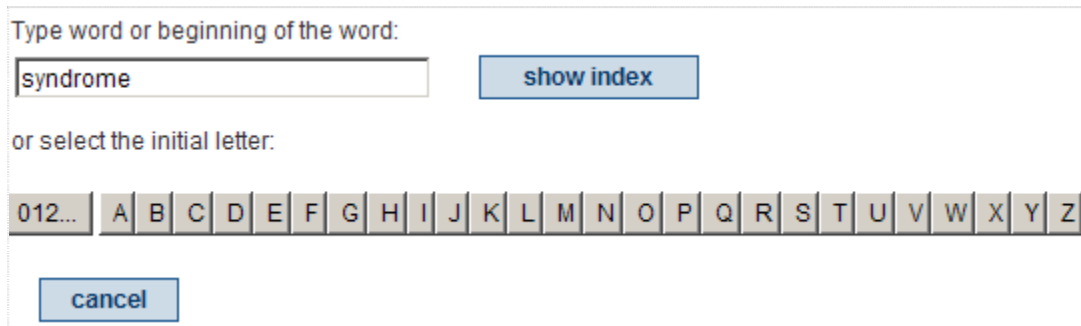


Figure 10: Example of how to complete the page for access to the index.

Step 3 – from the index display, click to mark the descriptor of interest

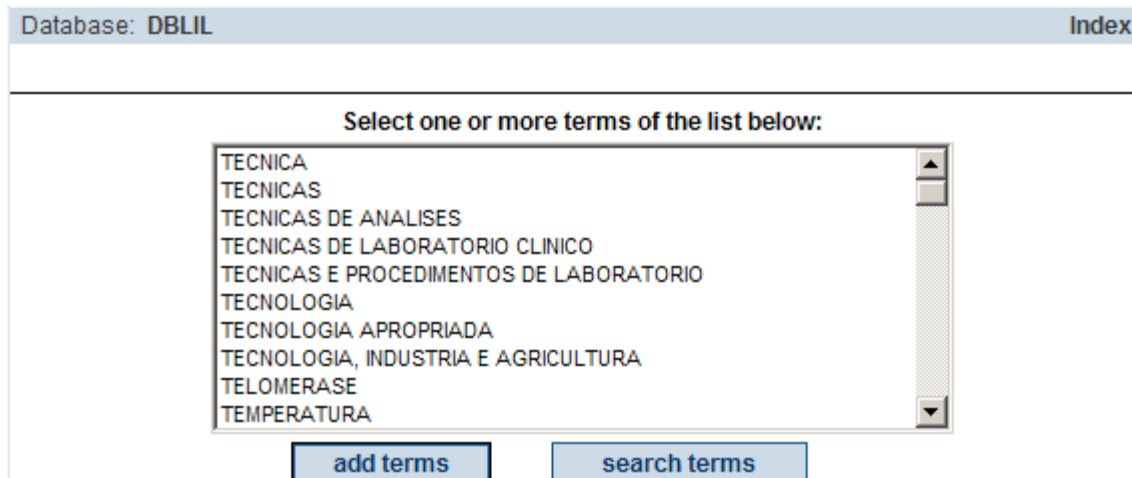
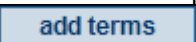


Figure 11: Example of the list of index terms.

1. To execute the search, click the button **search terms**.

The system retrieves all the records which contain the **descriptor** "alcoholic foetal syndrome".

2. To relate this descriptor with another descriptor or with another search field (language, author, etc.) click the button .

The system transfers the descriptor selected to the search form.

Appendix A - Indexing Techniques

For IAH to carry out a search in an ISIS database, it is necessary to perform an inversion of the data using an FST file (Field Select Table). FST files consist of one or more lines divided into three segments:

1. a field identifier (tag)
2. an indexing technique
3. a format for extraction of data, in the CDS/ISIS formatting language

The indexing techniques define the processes which are performed on the data generated by the format.

CDS/ISIS offers nine indexing techniques, identified by a single-digit code, as explained below:

- Technique 0 (zero)

Constructs a term from each line extracted by the format. This technique is generally used for indexing internal fields or subfields. However, note that CDS/ISIS constructs terms from lines and not fields. So you should realise that CDS/ISIS considers the output of the format as a string of characters in which the fields are no longer identifiable.

Therefore, it is the responsibility of the user to produce the correct data through the format, especially when indexing repeatable fields and/or more than one field. In other words, when using this technique, verify that the extraction format for data produces as output a line for each term that should be indexed.

- Technique 1

Constructs a term from each subfield or line extracted by the format considering delimiters. Since CDS/ISIS looks for subfield delimiters in the output of the format for the technique to function correctly, the format should specify **proof mode** (or no mode, since proof is the default) which retains the subfield delimiters in the output.

Recall that the modes "head" and "data" substitute the subfield delimiters with punctuation marks. Observe also that technique 1 is an extension of technique 0.

- Technique 2

Constructs a term from each term or phrase included between < and > (less than and greater than). It does not index any text except between the characters < and >. Note that this technique requires the **proof mode** because the other modes remove the characters < and >.

E.g.

"Report describing a <university course> on <training in documentation> in an African <library school>" produces the following index terms with this technique:

- university course
- training in documentation
- library school

- Technique 3

Executes the same processing as technique 2, except that the terms are included between slashes (/ ... /).

E.g.

"Report describing a /university course/ on /training in documentation/ in an African /library school/" produces the following index terms with this technique: university course

- training in documentation
- library school

- Technique 4

Constructs a term from each word of the text extracted by the format. By "a word" is meant a continuous sequence of alphabetic characters.

Observe that when you use this technique to index a database with subfield delimiters, you should specify the "head" or "data" mode (mhl or mdl) in the corresponding format for substituting the subfield delimiter according to the rules of that mode. Otherwise, the alphabetic codes of the subfield are considered as part of a word.

Also it is advisable to use the "head" or "data" mode if the field that will be indexed may contain file information, in case the form of presentation of the field is ignored.

Note: The definition of "alphabetic characters" can be adapted to the needs of each user by means of the System Table ISISAC.TAB.

- Techniques 5, 6, 7 and 8

Indexing techniques 5, 6, 7 and 8 operate in a similar way to techniques 1, 2, 3 and 4, respectively, but differ in the application of prefixes to the extracted terms.

The specification of a prefix is an unconditional literal using the general form:

'dprefixd', <CDS/ISIS format>

in which:

d	is a delimiter character not used as part of the prefix
prefix	is the string of characters
<CDS/ISIS format>	is the extraction format for data in CDS/ISIS

For example:

To prefix the words of the field 24 with "TI_", you should load an FST containing the following declaration:

24 8 '#TI_#',v24

in which:

Format	Description
'#TI_#'	unconditional literal
#	delimiter of the prefix
TI_	prefix to be used
v24	CDS/ISIS extraction format

Table 8: Example of extraction format from fields in the FST

For the sake of comparison, we may observe an FST containing extraction rules with and without the use of prefixes:

Without prefix	With prefix
70 0 mhu, (v70/)	70 0 mhu, ('AU_',v70/)
24 4 mhu, v24	24 8 mhu, '/TI_/',v24
69 2 v69	69 6 '/KW_/',v69

Table 9: Comparison of indexing techniques

It is important to note that the technique 0 (zero) is unique in that the use of a prefix is an option. Table 4 shows the relation between the different indexing techniques and the use of prefixes.

Without prefix	With prefix	Effect of the technique
0	0	Whole field
1	5	Subfield
2	6	Terms delimited by < and >
3	7	Terms delimited by /and /
4	8	Word by word

Short example of an FST with prefixes for a database compatible with the LILACS format

```
06 0 ( |TW_|v06/)
12 8 (" |TW_| "d12,v12|%|/)
13 8 (" |TW_| "d13,v13|%|/)
18 8 (" |TW_| "d18,v18|%|/)
19 8 (" |TW_| "d19,v19|%|/)
25 8 (" |TW_| "d25,v25|%|/)
12 8 (" |TI_| "d12,v12|%|/)
13 8 (" |TI_| "d13,v13|%|/)
18 8 (" |TI_| "d18,v18|%|/)
19 8 (" |TI_| "d19,,v19|%|/)
25 8 (" |TI_| "d25,v25|%|/)
40 5 (" |LA_| "d40,v40|%|/)
06 0 ( |NB_|v06/)
```

Short example of definitions (<base>.def file) compatible with the FST with prefixes

```
INDEX Tw=^1Palavras^2Palabras^3Words^d*^xTW ^uTW_^yDATABASE^mTW_
INDEX Ti=^1Palavras do título^2Palabras of the título^3Title words^xTI
^uTI_^yDATABASE^mTI_
INDEX La=^1Language^2Language^3Language^xLA ^uLA_^yDATABASE^mLA_^tshort
INDEX Nb=^3Bibliographic level^2Nivel bibliografico^1Nível bibliografico^xNB
^uNB_^yDATABASE^mNB_^tshort^fa^gGIZMONB
```

Appendix B - Fields of the LILACS format

- [01] – Code of the Centre
- [02] – Identification Number
- [03] – Location of the Document

[04] - Database
[05] - Type of Literature
[06] - Level of Processing
[07] – Call Number
[08] – Electronic Medium
[10] – Personal Author (Analytic level)
[11] – Corporate Author (Analytic level)
[12] - Title (Analytic level)
[13] - Title Translated into English (Analytic level)
[14] - Pages (Analytic level)
[16] – Personal Author (Monograph level)
[17] – Corporate Author Institutional (Monograph level)
[18] - Title (Monograph level)
[19] - Title Translated into English (Monograph level)
[20] - Pages (Monograph level)
[21] - Volume (Monograph level)
[23] – Personal Author (Collection level)
[24] – Corporate Author (Collection level)
[25] - Title (Collection level)
[27] – Total Number Total of Volumes (Collection level)
[30] - Title (Series level)
[31] - Volume (Series level)
[32] - Number of Part (Series level)
[35] - ISSN
[38] – Descriptive Information
[40] - Language of the Text
[41] - Language of the Abstract
[50] - Thesis and Dissertation – Institution to which it is presented
[51] – Thesis and Disertation - Title of Degree
[52] - Event – Sponsoring Institution
[53] - Event - Name
[54] - Event - Date
[55] - Event – Normalized Date
[56] - Event - Town
[57] - Event - Country
[58] - Project – Sponsoring Institution
[59] - Proyect - Name
[60] - Project - Number
[61] - Noted
[62] - Editor
[63] - Edition
[64] - Date of Publication
[65] – Normalized Date
[66] – Town of Publication
[67] - Country of Publication
[68] - Symbol
[69] - ISBN

- [71] - Type of Publication
- [72] - Number Total of References
- [74] – Date Range (from)
- [75] - Date Range (to)
- [76] – Pre-codified Descriptor
- [78] – Person as Subject
- [82] – DeCS Region Number
- [83] - Abstract
- [84] - Date of Entry on the Database
- [87] – Primary Descriptor
- [88] – Secondary Descriptor
- [91] - Date of processing
- [92] - Documentalist
- [98] – Related Record (Monograph, Grey literature, Collection, Series or Thesis)
- [101] – Related Record (Event)
- [102] – Related Record (Project)
- [899] – Version of Software

Appendix C - Searchable fields of the LILACS demo database

Field	Description
TW	Words
TI	Words from the title
AB	Words from the abstract
AU	Author
AI	Corporate author
MH	Descriptor
CT	Limits
LA	Language
PD	Country, year of publication
TA	Journal
TL	Type of literature
NB	Bibliographic level
MS	Monograph in series
PT	Type of publication
CF	Conference
ID	Unique identifier
LO	Localization

Field	Description
FE	Electronic medium
DT	Date of entry

Appendix D - How to construct IAH URLs with parameters

It is possible to construct a URL which accesses directly the search function of IAH. Below is the description of how to construct manually a typical URL.

1. Start the URL with the name of your domain in which IAH is installed and the search command:

```
http://my-server/cgi-bin/wxis.exe/iah/?IsisScript=iah/iah.xis&nextAction=lnk& ...
```

2. Add to the URL the parameters for the database to be searched and the language of search:


```
...&base=BASE&lang=LANGUAGE&...
```

where **BASE** should be the name of the database configured during the installation, for example: LILACS, and **LANGUAGE** should be: pt (for Portuguese), es (for Spanish), en (for English), or fr (for French).

3. Specify the search expression:

```
...&exprSearch=SEARCH&indexSearch=[INDEX]&conectSearch=[OPERATOR]&....
```

where **SEARCH** should be a boolean expression and **INDEX** can contain the specification of the field (for example DE).

You can create a boolean search expression with a maximum of three different fields, for which you should use the parameter **conectsearch** and state in **OPERATOR** one of the boolean operators available (AND, OR or AND NOT).

4. To search for the terms "health" and "public" in the MARC database:

```
nextAction=lnk&base=MARC&lang=pt&exprSearch=health+AND+public
```

5. To search the term "malaria" in the field of the subject headings in the DBLIL database:

```
nextAction=lnk&base=DBLIL&lang=en&exprSearch=lycopersicon+esculentum&indexSearch=MH
```

6. To search for "genetics" in the field of words and "English" in the language field in the DBLIL database:

```
nextAction=lnk&base=DBLIL&lang=en&exprSearch=genetics&indexSearch=TW&conectSearch=AND&
exprSearch=en&indexSearch=LA
```

Appendix E - Fields of the interface

Tag	Subfield	Description
5000		environment variables

Tag	Subfield	Description
	^a	current action
	^b	path_database
	^c	path_cgi-bin
	^d	path_data
	^s	name of the script
	^p	alternative PATH_TRANSLATED
	^f	definition file for the alternative application
	^l	information about the local server
	^t	temporary directory
5001		nextAction
	^s	status
5002		path for the directory of images
5003		name of the database
	^*	definition file
	^d	drive for "cipchange"
	^n	New – current altered database for executing function LoadBaseDef
5004		function baseResubmit
5005		formats available
	^n	Logical name
	^p	description in Portuguese
	^e	description in Spanish
	^i	description in English
5006		appearance
	^c	colour of background of the body
	^i	colour of background of the image
	^t	colour of text of the body
	^l	colour of links in the text
	^b	colour of the navigation bar
	^e	e-mail address of the administrador of the interface
	^m	selector of multi-language (ON/OFF)
	^r	selector of sample of results in reverse mode (ON/OFF)
	^v	colour of visited links in the text
5007		function navBar (navigation bar)
5008		number of records noted in the search
5009		name of the display format
5010		help information
	^n	identification of the type of information (help/explanatory note)
	^v	name of the HTML file
5011		identifier of sensitive help
	^h	help for current status of the system
	^n	note for current status of the system
5012		gizmo database for conversion/decoding
	^g	conversion (gizmo)
	^d	decoding
5013		name of the user
	^*	code of the user
	^m	mfn in the user database
5014		forms available (F,B,A)
5015		direct access to a specified index

Tag	Subfield	Description
5018		databaseFeatures
5020		database of execution log
5021		language
5030		logo (image)
5031		logo (URL)
5040		head (image)
5041		head (URL)
6000		form
6001		variable conectSearch
6002		variable exprSearch
6003		variable indexSearch
	^l	line of the form
	^n	name of the index
	^x	prefix
	^y	inverted file
	^u	prefix not inverted
	^t	type (kwic,short)
	^g	gizmo (applied gizmo in the index terms)
	^s	list of tags for search
6099		list of indexes
6100		Index selected
	^l	line of the form
	^n	name of the index
	^x	prefix of the index
	^y	inverted file
	^u	prefix not inverted
	^t	type (kwic,short)
	^g	gizmo (applies to gizmo not index terms)
	^s	list of tags for search
6111		gizmoIndex tag 1
6112		gizmoIndex tag 2
6102		termsFromIndex
6121		variable kwicMode
6122		variable indexRoot
6205		variable pagesRange
6209		list of options
6210		Selected item from list
	^m	mfn
6211		list of results Hit-list
6300		options of XML
6301		table XML
6302		formato xmlPFT
6400		related data
	^m	mfn
7000		limit of the search (restriction of the expression)