MetaLib Users Guide

Version 1.2

Table of Contents

1 METALIB - AN INTRODUCTION 4

2 METALIB ENVIRONMENT 5

2.1 Accessing MetaLib's Web-Based Management Interface 5

2.2 MetaLib Libraries 6

2.3 MetaLib Directory Tree 7

3 METALIB RESOURCES 8

3.1 Information Resources Database 8

3.1.1 Adding a New Resource 8
3.1.2 FIND Resource 15
3.1.3 Browse List of Resources 15
3.1.4 Modifying the Pull-Down Menus for Pre-defined Values 16
3.1.5 Information Resources Database Indexes 17

3.2 MetaLib Resource Configuration 17

3.2.1 The General Tab 19
3.2.2 Search Routines 23
3.2.3 Find Routines 24
3.2.4 Scan Routines 25
3.2.5 Fix-Scan Routines 26
3.2.6 Conversion 26
3.2.7 Conversion Table 26
3.2.8 Open URL 31
3.2.9 Conversion Table 32
3.2.10 Applications 35

3.3 Z39_Gate Configuration 37

4 METALIB ACCESS MANAGEMENT – AUTHENTICATION & AUTHORIZATION 38

4.1 User Authentication 39

4.2 User Authorizations 39

4.3 Bypassing the Sign-In Page 41

5 METALIB USER INTERFACE SETUP 43

5.1 Helping the User to Locate Resources 43

5.1.1 Info Gateway Pre-defined Groups of Resources 43
5.1.2 Locate Resources 44
5.1.3 Resource List Names (GROUP_TEXT) 44
5.1.4 Resource Search Limit & Automatic Selection of Resources 46

5.2 Merging and De-Duplication of Sets 46
5.2.1 De-Duplication Algorithm 47

5.3 Contact 48

6 METALIB SYSTEM MAINTENANCE 50

6.1 MetaLib Background Processes 50
6.1.1 Monitoring MetaLib Processes 50
6.1.2 Starting MetaLib Processes 51
6.1.3 Apache, WWW servers, z39_gate, Z39_server 51
6.1.4 Configuring Additional WWW Servers 52
6.1.5 Logfiles 52

6.2 MetaLib Backup 52

6.3 MetaLib Maintenance Procedures 53

1 MetaLib - an Introduction

MetaLib is a information portal that provides access to the full spectrum of the resources available to the institution. MetaLib is powered by the **Universal Gateway,** which enables the user to search in resources with different search and communication protocols as well as diverse data formats. Search results are then displayed to users in a uniform way. Resources that cannot be accessed via the Universal Gateway can be linked to directly.

MetaLib has an integrated '**Information Resources Database**'. Every MetaLib resource is cataloged in the database, including a description of the resource, its subject content and coverage, as well as administrative aspects such as access information, address, phone number, and so on. The 'Information Gateway' resource pre-defined groupings and the 'Locate Resources' function are based on the Information Resources Database indexes. The database is also used to display information about resources to the user.

2 MetaLib Environment

In order to work with MetaLib, you need to become familiar with MetaLib's environment. If you deal mainly with MetaLib resource configuration, you need to work with MetaLib's Web-based Management Guide. Other aspects of the MetaLib setup (for example the HTML pages) are held in files or tables on the server and you need to learn MetaLib's directory tree.

2.1 Accessing MetaLib's Web-Based Management Interface

MetaLib has a web-based user interface for the administration of various aspects of the Metalib installation. To access the MetaLib management interface, follow these steps:

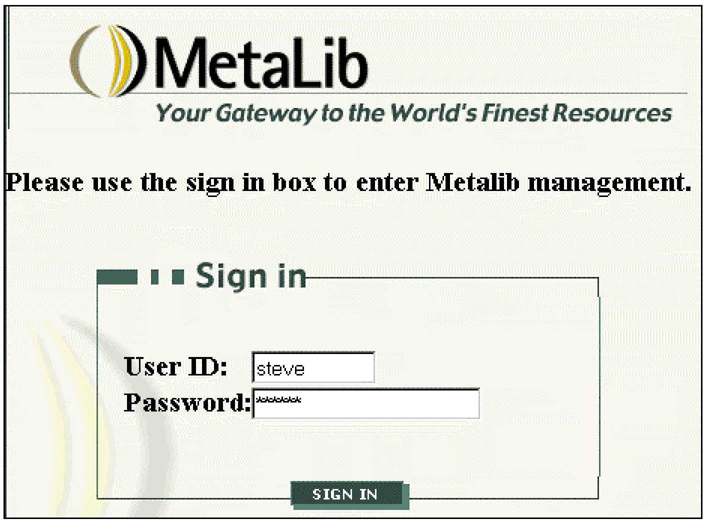
1. In your web browser, enter the same URL as MetaLib, but with an M after the slash rather than a V, as in the following:

   http://<IP or URL of MetaLib server>/M

   For example:

   http://www.metalib.com/M (note the letter M is capitalized)

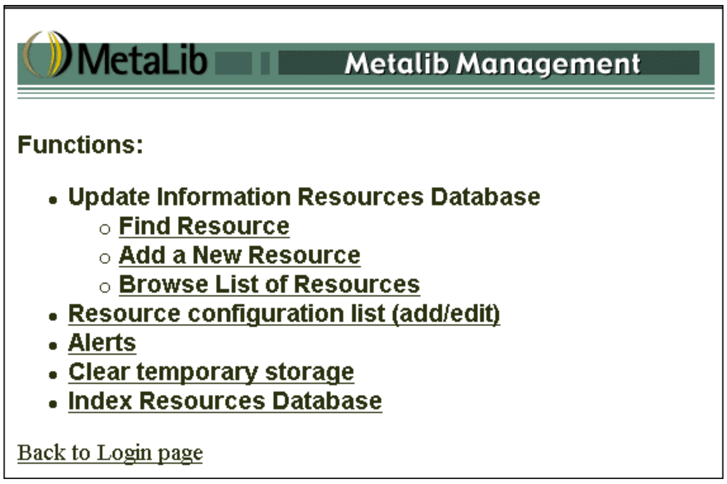
   The Sign In window appears:



Please use the sign in box to enter Metalib management.

User ID: steve
Password: ******

2. Type your user ID and Password and click the SIGN IN button. The MetaLib Management interface displays:


Metalib Management

Functions:

- Update Information Resources Database
  - Find Resource
  - Add a New Resource
  - Browse List of Resources
- Resource configuration list (add/edit)
- Alerts
- Clear temporary storage
- Index Resources Database

Back to Login page

The Management interface options are explained in the following sections of this guide.

❑ Update Information Resources Database	3.1
❑ Resource configuration list (add/edit)	3.2
❑ Clear Temporary Storage	6.3
❑ Index Resources Database	3.1.6

As we have seen, the web-based management interface is password-protected. Authorized users must be entered in the tab_management.lng table in $alephe_tab:

```
>> cd $alephe_tab
>> vi tab_management.lng
```

The structure of the table is as follows:

Col. 1	Username
Col. 2	Password
Col. 3	Allowed to use the web-based management interface – 'Y'(es)/'N'(o)

Example:

```
!  1                  2                     3
!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!|!|!|!
amit        amit                 Y
metalib     metalib              Y
```

2.2 MetaLib Libraries

When we refer to a "MetaLib Library", we mean an Oracle database with a set of configuration files and tables that serve as the basis of a single MetaLib installation.

Every MetaLib installation has three libraries, VIR00, VIR01 and DAT01.

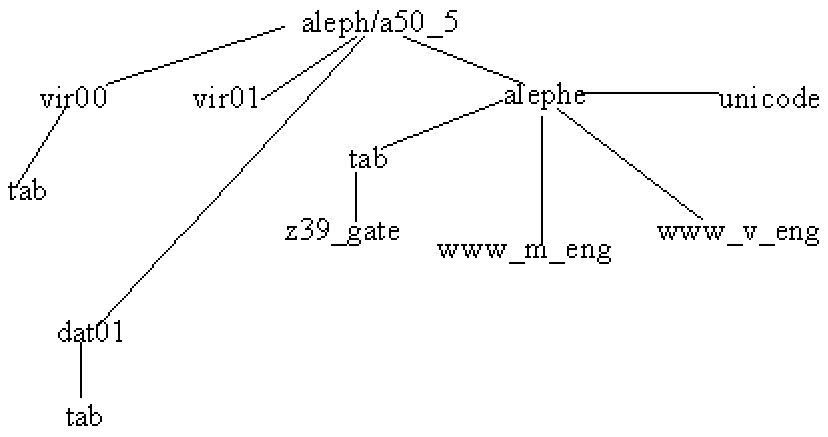
VIR00 is the “permanent” library. It stores data that must be retained over time, primarily the user profiles, as well as the MetaLib database setup tables and MetaLib resource configuration information.

VIR01 is the temporary database where MetaLib temporarily stores recordsretrieved by the Universal Gateway. This database must be deleted periodically (refer to section D.3) as it “grows” or “expands” quickly and may fill up the system.

DAT01 is the library used by the Information Resources Database.

2.3 MetaLib Directory Tree

A MetaLib installation has the following directory tree,



You can move to key directories with the following alias names (an alias serves in this case as a shortcut for moving to the relevant directory):

Full path	Alias	
/aleph/a50_5/vir00 /aleph/a50_5/vir01 /aleph/a50_5/dat01	Dlib vir00 Dlib vir01 Dlib dat01	root directory of the library
/aleph/a50_5/vir00/tab	dt	Library tables. Note that you can use this shortcut if you have moved already to the vir00 library, otherwise you must first enter dlib vir00
/aleph/a50_5/alephe	$alephe_root	root directory of alephe – has some basic set-up tables
/aleph/a50_5/alephe/tab/z39_gate	$alephe_tab/z39_gate	z39 configuration files
/aleph/a50_5alephe/www_v_eng	wv	MetaLib user interface HTML files
/aleph/a50_5alephe/www_m_eng	wm	MetaLib management interface HTML files
/aleph/a50_5alephe/unicode	$alephe_unicode	Character conversion files

3 MetaLib Resources

There are two kinds of MetaLib resources:

Resources that can be searched via the Universal Gateway. These resources can be searched together using the MetaLib search interface. If the user prefers, the resources can also be linked to directly. These resources are referred to as MetaLib search resources or MetaLib targets.

Resources that cannot be searched via the Universal Gateway. These resources can only be linked to directly and searched using the information provider's interface.

All resources must be cataloged in the Information Resources Database. A MetaLib search resource must also have a MetaLib configuration record. The following sections explain how to catalog resources in the Information Resources Database and how to create a resource configuration record.

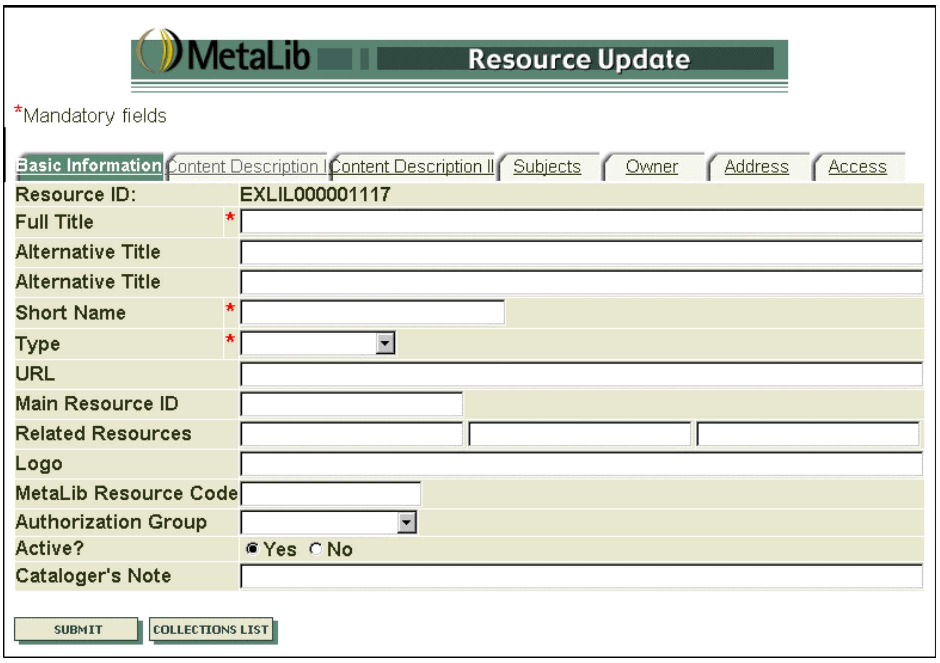
3.1 Information Resources Database

The first step in adding a new MetaLib resource is to catalog it in the Information Resources Database – or IRD. IRD records can be searched and cataloged from the MetaLib Web-based Management Interface. The following options are available:

- Add a New Resource
- Find a Resource
- Browse List of Resources

3.1.1 Adding a New Resource

When the "Add a New Resource" option is invoked from the Management window, the system displays a form for entering information about the resource:



*Mandatory fields

Resource ID:	EXLIL000001117
Full Title *	
Alternative Title	
Alternative Title	
Short Name *	
Type *	
URL	
Main Resource ID	
Related Resources	
Logo	
MetaLib Resource Code	
Authorization Group	
Active?	[✓ Yes] No
Cataloger's Note	

As you can see, the form is arranged in tabs.

The following fields are defined in the Information Resources Database. It is not mandatory to fill in every field. Fields which are mandatory are labeled as such.

Tab A: Basic Information:

Resource ID
System assigned unique record ID. The ID consists of a prefix which indicates where the record was created; digital Resources created by Ex Libris are prefixed by 'EXLIL'. Locally created resources have a different prefix.

Full Title (Mandatory)
The title or full name of the resource. (This field is the equivalent of the RESOURCE-FULLNAME in the MetaLib resource configuration file)
Maximum length: 100 characters

Alternative Title
Alternative titles or names by which the resource is known.
Maximum length: 200 characters

Short Name (Mandatory)
A brief title or name of the resource. This is the name that displays througout MetaLib. This field must be the same as the RESOURCE-SHORTNAME in the MetaLib resource configuration file.
Maximum length: 30 characters

Type (Mandatory)
Resource type. The following categories have been defined (additional types can be added locally. Refer to section 3.1.5)

- University Library
- Public Library
- Special Library
- National Library
- Corporate Library
- Image Library
- Database
- Archive
- Search Engine
- Newspaper
- Museum
- Journal Collection

URL
The URL of the resource for direct access. Resources with a URL will be underlined in MetaLib to enable a direct link.
Maximum length: 200 characters

Main Resource ID & Related Resources

These fields are used to create links and a logical relationship between resources. There are two types of links:

1) Hierarchical. This relationship is entered in the “Main Resource ID” field.
2) Parallel. This relationship is entered in the “Other related Resource” field/s

In a hierarchical relationship, one resource is part of a larger resource. For example, if you are cataloging a special archive within a university library, then the Main Resource ID would be the university library. Note that only UP links to the main resource are entered from the secondary resource/s. The system displays the link from both the main and the secondary resources.

In a parallel-type relationship, resources are independent of each other but are related in some way. The link is used to inform the end-user of the existence of such resources. For example, you might want to link two resources on the same topic. It is sufficient to enter the link on one of the records; the system displays the link from both resources in the user interface.

In both cases enter the Resources ID (for example, EXL0000000109).

[Note: this linking mechanism is still in development]

Logo

This field may be used for entering a URL to the resource’s or owner’s logo.
Maximum length: 200 characters

MetaLib Resource Code (mandatory when the resource can be searched via MetaLib Search)

If the resource can be searched via MetaLib Search (that is, it has a resource configuration file, Z58), enter the MetaLib RESOURCE code in this field.

→ Note:

A MetaLib resource code may be entered also for resources that are only subsets of a MetaLib search resource. In this case, MetaLib displays the name of the subset resource and the name of the main resource. For example, the Einstein Collection is not a MetaLib search resource. However, it is a subset of the Hebrew University’s National Library which is a MetaLib search resource. If the MetaLib Resource code of the National Library is entered in the ‘MetaLib Resource Code’ field of the Einstein Collection’s IRD record, it is treated as a MetaLib search resource. In MetaLib, the Einstein Collection displays as “Einstein Collection (HU National Library)” to indicate that it is only a subset of the resource actually searched.

Maximum length: 20 characters

Authorization Group

The authorization group can be used to group resources for access control purposes – that is, to decide which users can search which resources. Access control parameters are defined in a table (see section 4.2). Examples of authorization groups can be "Free" or "Restricted". In a institution with campuses that subscribe to different resources, resources can be grouped by campus. Note that every resource can belong to more than one group.

The list of authorization groups can be modified locally. Refer to section 3.1.5.

Maximum length: 10 characters

Active (Y/N)

The "active/non-active" status enables the institution to control which resources are offered to the user**.** Only "active" resources are seen by the end-user.

Note that the active status can also be updated from the "Browse List of Resources" function.

Cataloger's note

This field can be used to record any information about the cataloger and the time the record was cataloged.
Maximum length: 200 characters

Tab B: Content Description I

Info Gateway Groups

This field is used to link a resource to one or more of the "Information Gateway's" pre-defined groupings of resources (for example "Art"). This list can be modified locally. Refer to section 3.1.5.

Description

Use this field to describe the resource in general terms.
Maximum length: 1900 characters

OPAC Description

This field is used to describe the OPAC of a resource. It is advisable to use this field if the resources' OPAC reflects only part of the full collection.
Maximum length: 1900 characters

Tab C: Content Description II

Coverage

Use this field to describe the extent or scope of the resource. Coverage typically includes geographic scope (Egypt) or temporal period (19th Century).
Maximum length: 500 characters

Purpose
Use this field to explain why the resource was created; its origins and its importance. The field can also be used to describe resource policy.
Maximum length: 500 characters

Languages
The main language/s of the content of the resource.

Notes
The note fields can be used to enter any specialized information that cannot be entered in one of the other fields.
Maximum length: 200 characters

Tab D: Subjects

Controlled Subjects
The main broad topic/s covered by the resource. Select from the **pre-defined list** which can be customized locally. Refer to section 3.1.5.

Keywords
Additional, free-text, keywords to describe the resource's content. Keywords can be more specific and should not repeat the controlled subjects.
Maximum length: 50 characters

Tab E: Owner

Owner
Enter the resource owner, the person or organization that owns the resource. This field does not have to be entered if the Owner's name is identical to the name of the resource.
Maximum length: 200 characters

Type
Enter the type or category of the owner in this field. Select from one of the following These types can be modified locally (refer to section 3.1.5).

- ❑ University
- ❑ Company
- ❑ Database Vendor
- ❑ Government Agency
- ❑ Municipality

Owner Alternative Names
Additional owner names.
Maximum length: 200 characters

Owner ID
Identifier of the owner if available. For example, a national identifier.
Maximum length: 25 characters

Owner Description
A general description of the owner.
Maximum length: 1000 characters

Owner URL
URL to the homepage of the owner (if it differs from the URL of the resource).
Maximum length: 200 characters

Publisher
The publisher is the entity responsible for making the resource available in its present form. The publisher is very often the same as the owner.
Maximum length: 200 characters

Services
Use this field to record any services offered by the resource or the owner of the resource.
Maximum length: 250 characters

Administrator
Use this field to enter information concerning the person who administers the resource (name and email).
Maximum length: 200 characters

Tab F: Address

Fields in this tab are used to enter the resource's address (if this is relevant.)

Address
Use this field to enter the resource's street and number.
Maximum length: 200 characters

City
Use this field to enter the city name.
Maximum length: 30 characters

State/Province
Use this field to enter the resource's state or province.
Maximum length: 30 characters

Postal code
Use this field to enter the resource's postal code
Maximum length: 30 characters

Country
Use this field to enter the country.

Maximum length: 30 characters

Telephone
Use these fields to enter telephone number/s.
Maximum length: 30 characters

Fax
Use these fields to enter fax number/s.
Maximum length: 30 characters

Email
Use this field to enter an email for the resource.
Maximum length: 100 characters

Contact
Use this field to enter the name of a contact person.
Maximum length: 200 characters

Note
Use this field for any information that could not be entered in another field.
Maximum length: 200 characters

Transportation
Use this field to describe the transportation available where relevant.
Maximum length: 1000 characters

Access for disabled
Use this field to describe special access for the disabled, when relevant.
Maximum length: 250 characters

Location code
Use this field to enter a national or regional location code. This code may serve for the creation of an HTML Web map of the resources in an area.
Maximum length: 50 characters

Tab G: Access

Access Policy
Use this field to describe who may use the resource and whether there are any limitations or pre-requisites for use.
Maximum length: 200 characters

Charging (loaning) policy
A description of any payment mechanism necessary for accessing the resource.
Maximum length: 200 characters

Opening hours

Use these fields to describe the opening hours of the resource.
Maximum length: 150 characters per field

Opening hours - note

Maximum length: 150 characters

Opening hours - URL

Link to a page describing the opening hours.
Maximum length: 200

3.1.2 FIND Resource

Use this function to locate resources for update purposes. The following IRD record fields can be searched:

- Resource ID Number
- Resource name – searches in full, short and alternative names
- MetaLib code
- Info Gateway Groups
- Resource Type
- Owner
- Subject
- Any words – searches in all resource names, resource type, controlled subjects, keywords, languages, description, OPAC description, address, owner names, owner services, notes
- Status – by default searches all resources; the search can be limited to active or non-active resources
- System number – search by internal system number

The resources located in the search are displayed in a 'Resource List'. The functions available from this screen are explained below.

Note that up to 100 records can be located in a search. If your query finds more than 100 records, you are asked to refine your search.

3.1.3 Browse List of Resources

This function enables you to browse a list of resource names (the full name is displayed). From this list, you can update the status of the resource (active=checked; non-active=no check). Clicking on the UPDATE button displays the Resource list and the functions for modifying it.

These functions are explained below.

List of Resources

The following functions are available from the 'Resource List'

Modify	displays the IRD record form for update purposes
New	displays a form for entering a new IRD record
Copy	uses a selected IRD record as the basis for creating a new record
Delete	deletes an IRD record
Find	displays the FIND resource screen
Main menu	returns to MetaLib management menu

3.1.4 Modifying the Pull-Down Menus for Pre-defined Values

Some of the IRD fields are selected from pre-defined lists of values and can be modified locally. In particular, the values for the "Authorization groups" and the "Info Gateway Groups" are generally locally defined.

The pre-defined values are defined in the HTML files of the IRD cataloging interface. The files are in the directory 'www_m_lng":

```
>> cd $alephe_root
>> cd www_m_lng

        OR just

>> wm
```

The following files must be modified:

1) Authorization Group

HTML file: doc-a

Example:

```
<tr>
    <td class=td1 id=smallb width=15% colspan=2 nowrap>
      Authorization Group
    </td>
    <td class=td1 id=smallb colspan=3>
      <select name="09">
        <option value=""></option>
        <option value="FREE" $$0900-S"FREE">Free Resources</option>
        <option    value="RESTRICT"    $$0900-S"RESTRICT">Restricted
resources</option
>
      </select>
    </td>
```

→ **Note**: the length of the field in the database, the content of "option value", is limited to 10 characters. This is the authorization group that is entered in the 'tab_authorize_source' table. Refer to section 4.2.

2) Info Gateway Groups

HTML file - doc-b

3) Resource type

HTML file: doc-a

→ **Note**: if the resource types in the cataloging interface/module are changed the list should also be changed in the 'Locate Resources' function (HTML file:www_v_eng/dod-type)

4) Controlled subjects

HTML file: doc-d

→ **Note**: if the controlled subjects in the cataloging interface are changed, the list should also be changed in the 'Locate Resources' function (HTML file:www_v_eng/dod-collection)

5) Owner type

HTML file: doc-e

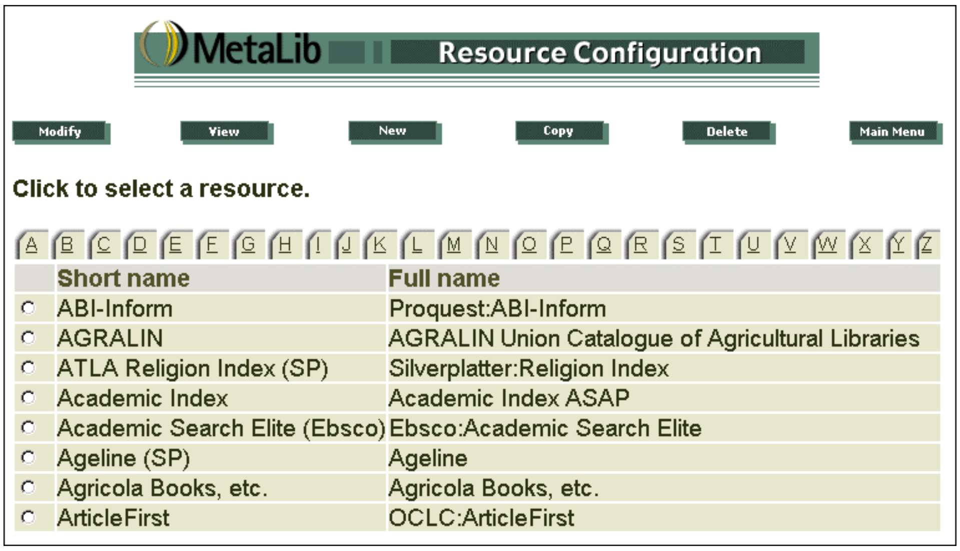
3.1.5 Information Resources Database Indexes

The IRD indexes are created automatically when the records are saved to the database. Occasionally, it may be necessary to rebuild the indexes from scratch. In this case, use the 'Index Resources Database' facility/option from the MetaLib Management interface.

3.2 MetaLib Resource Configuration

MetaLib search accesses databases and catalogs, using a variety of protocols. Configuration files and conversion programs convert a MetaLib query to the format suitable to the MetaLib resource target database, and then convert the retrieved records to a uniform format – MARC 21— using the Unicode (UTF-8) character code.

MetaLib resources are configured via the web-based MetaLib Management interface. (http://<IP>:port/M . Click on 'Resource configuration list' to display a list of resources that are arranged in alphabetic tabs:


MetaLib Resource Configuration
Modify
View
New
Copy
Delete
Main Menu
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Click to select a resource.

	Short name	Full name
○	ABI-Inform	Proquest:ABI-Inform
○	AGRALIN	AGRALIN Union Catalogue of Agricultural Libraries
○	ATLA Religion Index (SP)	Silverplatter:Religion Index
○	Academic Index	Academic Index ASAP
○	Academic Search Elite (Ebsco)	Ebsco:Academic Search Elite
○	Ageline (SP)	Ageline
○	Agricola Books, etc.	Agricola Books, etc.
○	ArticleFirst	OCLC:ArticleFirst

The following options are available:

- Modify – update resource configuration information.
- View – view resource configuration information.
- New – add a new resource.
- Copy – add a new resource based on the configuration of the selected resource.
- Delete – delete resource.
- Main menu – return to the main menu of the MetaLib Management Interface.

For each resource, five tabs of information need to be filled in:

1) **General**. This tab defines basic aspects of the MetaLib resource for the Universal Gateway, including naming conventions, format and character set conversions

2) **Search Routines**. This tab form defines transformations necessary in order to adapt the query to the various search options in MetaLib for each MetaLib resource.

3) **Conversion**. This tab defines the conversion routines or conversion table for converting retrieved records from the target's original format to the uniform MetaLib format.

4) **OpenURL.** This tab defines the fields and programs, to be used for the creation of the Open URL for each resource. The OpenURL is created in order to provide SFX services.

5) **Applications.** This tab defines information needed by the various applications that utilize the Universal Gateway.

Note that in the resource list, two resources are defined for internal purposes and should not be deleted.   They are as follows: .

1) MERGESET - the RESOURCE-SHORTNAME, 'Merged Set' is used as the header for merged sets.

2) SELECT - the RESOURCE-SHORTNAME, "Selected" is used as the header for the set of selected records.

These resources require only the RESOURCE and RESOURCE-SHORTNAME.

3.2.1 The General Tab

Note: Not all fields must be filled out. Those which must be entered, are marked as "mandatory" below.

The 'General' tab has the following definitions:

RESOURCE (Mandatory)
This is the code of the resource as it has been defined in MetaLib. This code should be used in the "MetaLib resource code" field in the Information Resources Database. If the database is one of several databases from the same vendor, the RESOURCE should be in the form 'vendor_database. For instance, all EBSCO databases will be in the form of EBSCO_AFH, EBSCO_ERIC, EBSCO_SOCIO.

The code should be in upper case and there should be no spaces.

RESOURCE-FULLNAME
This is the full name of the resource. This name displays in the resource configuration list in the Web Management interface and in the info windows .

Example:
Ebsco:Sociological Abstracts

RESOURCE-SHORTNAME (Mandatory)
This is the brief name of the resource. It is limited to 30 characters. This is the resource name that displays througout MetaLib's user interface.

Example :
Sociological Abstracts

ACCESS_METHOD (Mandatory)
This is the protocol that is used to access the resource. At present, the following access methods are available:

Protocol	**Access-method code**
Z39.50	Z39
HTTP	Consult with MetaLib support
ALEPH	For ALEPH300 useALEPH300

	For ALEPH500 11.4 useALEPH_11_4 For ALEPH500 12.1 use ALEPH_12_1 For ALEPH500 12.2 use ALEPH_12_2 For ALEPH500 14.2 use ALEPH
External program using MetaLib's hook (refer to separate documentation)	EXTERNAL

HOST-NAME (Mandatory except for Z39 resources)
This is the address of the server and the port number where the database can be accessed. For Z39 resources, this information is entered in the Z39 configuration file of the resource in $alephe_tab/z39_gate.

Example:
se2.ub.fu-berlin.de:6505

DATABASE (Mandatory)

For Z39 targets:
The code entered here provides the link to the Z39 configuration file. The 'DATABASE' field must be the same (including case) as the resource code in the z39_gate_<RESOURCE>.conf file in $alephe_tab/z39_gate.

It is simplest if this 'DATABASE' is the same as 'RESOURCE'.

For example:

RESOURCE:	SILVER_ERIC
RESOURCE-FULLNAME	SilverPlatter:ERIC
RESOURCE-SHORTNAME	ERIC (SP)
DATABASE	SILVER_ERIC

The z39_gate configuration file must be: z39_gate_SILVER_ERIC.conf

For non-Z39 targets:
This is the code of the database as defined by the MetaLib target. This code must be requested from the provider of the database. The 'DATABASE' should be entered in upper case.

Example:
TUB01

IN-RECORD-TYPE (Mandatory)
This parameter defines the data format of the record retrieved by MetaLib from the target database. At present, MetaLib supports the following formats:

- USMARC
- UNIMARC

- DANMARC
- MAB
- SUTRS
- ALEPH300 (for ALEPH300 databases)

Example:
    USMARC

IN-RECORD-CREATE (Mandatory)
This parameter defines the program used for converting the incoming records into standard MetaLib format. The default program is vir_z00_z39_usmarc for ISO 2709 format.

The following programs are available for standard formats"
    vir_z00_z39_mab - for MAB format
    vir_z00_z39_usmarc – for ISO 2709 (MARC) format

There are some additional programs for non-standard formats.

IN-RECORD-CREATE-PARAM
Not in use

IN-RECORD-CHAR-CONV (Mandatory)
MetaLib converts all records to Unicode (UTF8). This parameter defines the character conversion routine to be used for the target database. The following are the available character conversion procedures:

ANSEL_TO_UTF
MAB_TO_UTF
8859_1_TO_UTF
8859_5_TO_UTF
8859_7_TO_UTF
8859_8_TO_UTF
LABT_TO_UTF – for Lithuanian ALEPH500 libraries.

IN-DIRECT-KEY & IN-DIRECT-KEY-PARAM
These parameters define which data element from the source record is used to create the SID field. The SID field is the basis for creating a direct link from the record displayed in MetaLib to the record in the target database (see WEB-ACCESS-DIRECT explained above).

There are two possible values for IN-DIRECT-KEY:

TAG	indicates that a field from the source record is used
SYSNO	indicates that the system number is used. The SYSNO parameter can be used for resources using the ALEPH protocol only.

The IN-DIRECT-KEY-PARAM defines which field from the source record is used. This parameter is relevant only if IN-DIRECT-KEY is TAG. This field should be the ID of the record in the target database.

The SID field has the following sub-fields:

a	access method
b	resource code
c	ID of the record in the target database (the content of the field defined in IN-DIRECT-KEY-PARAM or the ALEPH system number). The content of sub-field c is inserted in place of $0100 in the link-to syntax defined in the WEB-ACCESS-DIRECT parameter.

Example:

```
IN-DIRECT-KEY            TAG
IN-DIRECT-KEY-PARAM      001
```

IN-SCAN-FIX
Not in use

IN-SCAN-CHAR-CONV
This parameter defines which character conversion routine is used for the conversion of source records to UTF-8 in a ***Browse*** type search. The following routines are available:

ANSEL_TO_UTF
MAB_TO_UTF
8859_1_TO_UTF
8859_5_TO_UTF
8859_7_TO_UTF
8859_8_TO_UTF
LABT_TO_UTF (for Lithuanian ALEPH500 libraries)

OUT-FIND-CHAR-CONV (Mandatory)
This parameter defines the character conversion routine applied in a search query. MetaLib formulates queries in UTF-8 which must be converted to the character set required by the target database. The following conversion routines are available:

UTF_TO_8859_1
UTF_TO_8859_5
UTF_TO_8859_7
UTF_TO_8859_8
UTF_TO_MAB
UTF_TO_ANSEL
UTF_TO_LABT (for Lithuanian ALEPH libraries)

OUT-SCAN-CHAR-CONV
This parameter defines the character conversion applied in a Browse query. MetaLib formulates queries in UTF-8 which must be converted to the character set required by the target database. The following conversion routines are available:

UTF_TO_8859_1
UTF_TO_8859_5
UTF_TO_8859_7
UTF_TO_8859_8
UTF_TO_ANSEL
UTF_TO_LABT (for Lithuanian ALEPH libraries)
UTF_TO_MAB

SORT (Y/N)
This parameter defines whether or not a resource supports sorting of result sets. If a 'Y' is entered, the headers in the results list are hyperlinked enabling the user to click on them and sort according to the specific field (Author, Title, Year). When the caption is clicked, MetaLib sends a request to the target database to sort the results list. (Refer also to section B.4.2).

SCAN (Y/N)
Not in use.

3.2.2 Search Routines

The 'Search Routines' tab defines routines or transformations which are necessary in order to adapt the query to the various search options in MetaLib for each MetaLib resource.

There are three search/browse options:

Find	defines the routines required for transmitting a search from the main MetaLib search screen and from the **full** display of a record (search based on the author, title, or subject of the record).
Scan	defines the routines required for transmitting the search from the main MetaLib Index screen.
Fix-scan	MetaLib creates a merged index list from several target databases. This parameter defines routines to standardize the incoming index record. When a user clicks on a index record, MetaLib launches a new search using the text of the selected record. This section also defines which Find routines are used for the new search.

For every routine the following elements are defined:

Type
Find/Scan/Fix-scan

Code

These are the MetaLib codes. MetaLib codes are based on CCL.

Target Code

Translation of Metalib code to target database code. For Z39 databases this must also be defined in the z39_gate configuration.

Transformations

Term transformations consist of a sequence of small changes to be performed sequentially on the search term so that it is suitable for searching the target. For example, changing the search string to lowercase.

Each transformation has a number. Transformations are divided by an underscore. A transformation may have parameters in which a 'P' is entered after the transformation number, followed by the parameters and then an additional 'P".

See attached document 'term_transformation' for a description of the available transformations.

Use Sub-fields

Defines which sub-fields should be used. This element is used only if it is necessary to limit the sub-fields. This is not relevant to all search routines.

3.2.3 Find Routines

Find routines should be defined for each search option offered from the main Search screen:

- All fields (WRD)
- Authors (WAU)
- Titles (WTI)
- Subjects (WSU)

For example:

Type	Code	Target code	Transformations	Sub-fields
Find	WAU	WPE	13P<<a>>P_7P([/<P_2P.*P	

In the example above, the MetaLib code 'WAU' is translated to the target's Code, WPE. Sub-fields are not relevant in in this type of transformation.

<u>Note:</u>

If a target does not have a specific word file (for example, it does not have subjects), you can redirect the search to the general word file.

For example:

Type	Code	Target code	Transformations	Sub-fields
Find	WSU	WRD	13P<<a>>P_7P([/<P_2P.,!;:"-*P	

You must also define Find transformation for each field from which a search can be launched from the full record display:

1#### - for author search
7#### - for author search
24### - for title search
6#### - for subject search

These transformations normalize the text and create queries which are then passed through the find transformation defined for the search type (as defined in the transformation) for each target database.

For example:

Type	Code	Target code	Transformations	Sub-fields
Find	1####	WAU	1_7P([/<P	a

This means that when a search is launched from an author in record (MARC21 100, 110, 111, and so on), MetaLib takes only sub-field a from the author, normalizes the text for the query and then sends it to each target using the WAU transformations (as if the user typed it in the main search screen).

3.2.4 Scan Routines

Scan routines must be defined for each Browse option:

- Authors (AUT)
- Titles (TIT)
- Subjects (SUB)

For example:

Type	Code	Target code	Transformations	Sub-fields
Scan	AUT	PER	13P<<a>>P_7P([/<P_2P*P	

In the example above, the MetaLib code 'AUT' is translated to the target's code, 'PER'.

3.2.5 Fix-Scan Routines

Fix-scan routines must be defined for each browse option:

- Authors (AUT)
- Titles (TIT)
- Subjects (SUB)

Example:

Type	Code	Target code	Transformations	Sub-fields
Fix-scan	AUT	WAU	r=1_13P<<a>>P	ab

The example above normalizes the text for the author's browse list, taking only sub-fields a and b from the source record. The 'target code' defines which find routine is to be used when a search is launched from a browse list.
In this example, the find routine defined for WAU is used.

3.2.6 Conversion

MetaLib converts all record formats to MARC21 for consistent display but retains the original record format. In the 'Converion' tab, the fields are mapped from the original format to the MARC21 format. In addition to converting fields, the librarian can delete, merge or create new fields by extracting data from the original format. This conversion is applied before the record is displayed.

Conversion program
The default conversion program is called vir_fix_doc_standard'. In special cases, it may be necessary to write a special program.

Note: In the 'Conversion' tab it is possible to define up to 40 fields for conversion. If you need more, enter 'vir_fix_doc_table' in the conversion program field. In this case, the conversion table is created on the server in the directory $alephe_tab. The name of the table must be tab_conv_<MetaLib resource code in lower case>. For example, if the MetaLib resource code is MADISON the name of the conversion table should be:

```
tab_conv_madison
```

The table has the same columns as the 'Conversion' tab in the management interface.

Conversion parameters
Parameters for a special conversion program.

3.2.7 Conversion Table

The following columns are defined:

Col.1 In Tag
This is the tag of the field in the original format. Tags are limited to 5 characters.

Number signs (#) can be used as wildcards.

Col.2 In Sub Field
This is the subfield/s to be used from the field in the original format. Up to 10 subfields of one character can be defined.

Col.3 Program
If the incoming text needs to be processed, the conversion cannot simply change the tag and subfields - special programs can be used. The parameters for the program are entereded in the 'Param' column At present, the following programs are available:

vir_fix_doc_first
Defines initial position and length of string to take from the beginning of the tag/subfield. Note that the count starts from position 1.

vir_fix_doc_last
Defines initial position and length of string to take from the end of the tag/subfield. Note that the count starts from position 0.

vir_fix_doc_parser
Parser program to extract text from the middle of tag/subfield.

vir_fix_doc_is_zetoc
A special program written for Zetoc to distinguish the issn from the isbn. Since the original field, IS, can contain either, the program says that if there are 10 digits in the field, convert to 020 (isbn) and if there are 8 digits, convert to 022 (issn).

vir_fix_doc_aster
A special program to remove an asterisk appearing before a subject term, so that it is not sent to the SFX server (through the Open URL) with the asterisk.

Example from Pubmed:
```
In tag        In subfield     Program            Parameters      Action
650                a          vir_fix_doc_aster    650,a,           P
```

Note that only a P is used in the parameter to indicate a Program.

Col.4 Parameters
This column contains two elements:

1. The resulting tag and subfield/s.
2. Parameters for a program entered in the previous column.

Commas are used as delimiters.

The following types of parameters are possible:

vir_fix_doc_first
The parameters for this program are initial position and length of string. For example:

```
In tag       In sub       Program                  Param
008                       vir_fix_doc_first        YR,a,8,4
```

This means that 4 characters starting from the 8th position from tag 008 should be used to create the YR field, subfield a:

```
         ------>
         12345678
      008 000814s1975----dcu------b----00010-eng--
```

vir_fix_doc_last
The parameters for this program are initial position and length of string from the end of the tag/subfield. For example:

```
In tag       In sub       Program                  Param

510##        b            vir_fix_doc_last         YR,a,3,4
```

This means that the last four characters from subfield b of tag 510 should be used to create the YR field, subfield a:

```
                  3210
                  <---
      510   b  Jan, 1999
```

vir_fix_doc_parser
The parameters for this program are:

P - program parameter. Subfield to be used. This can be blank.
\S - indicates the start of parsing
\E - indicates the end of parsing. Use ‘E*’ to indicate that all text to the end of the line. E without an asterisk indicates that text to the first space or dash should be taken. Note that spaces do not have to be defined.

The parameters P, \S and \E must be used. Additional parameters:

\A - Used to indicate any alphabetic character. This parameter is used in conjunction with \S or \E to indicate the start or ending of parsing from the first alphabetic character.
\N - Used to indicate any numeric character. This parameter is used in conjunction with \S or \E to indicate the start or ending of parsing from the first numeric character.

For example:

```
In tag       In sub       Program                  Param
```

```
500         a           vir_fix_doc_parser      YR,a,Pa\SYear       of
Publication:\E
```

This means that the YR field, subfield a, should be created from text in subfield a of the 500 tag starting from "Year of Publication:" and until the first space (in this example also the end of the field).

> 500 |a Year of Publication: 1989

If the field has additional text after the year with no spaces for example:

> 500 |a Year of Publication: 1989newyork

the field could be parsed as follows:

```
In tag      In sub      Program                 Param
500         a           vir_fix_doc_parser      YR,a,Pa\SYear
ofPublication:\E\N
```

This means that the field is parsed from "Year of Publication" and until the first alphabetic character. Note that if there was a space between the year and the text it would have been sufficient to use '\E'.

vir_fix_doc_aster
The parameters for this program are the field, subfield and a "P" to indicate a program

Example from Pubmed:

```
In tag        In subfield     Program               Parameters

650            a        vir_fix_doc_aster     650,a,P
```

vir_fix_doc_zetoc
No parameters.

Example:

```
In tag        In subfield     Program               Parameters

IS####                        vir_fix_doc_zetoc
```

<u>Col.5 Action</u>
The 'action' parameter defines what should be done with the field. The following 'actions' are available:

> Blank (that is, no action defined) indicates that the field from the original format should be replaced by the converted field
>
> ADDNEW - indicates that the field should be added as a new field to the converted record, retaining the original field.

DELETE - indicates that the field should be deleted from the converted record. The DELETE action works on complete fields (that is, do not define it for subfields of the field).

SPLIT - splits a field with mutiple subfields into separate fields for each subfield.

MERGE - merges several fields into one field with subfields. See example below.

Note that the system displays all tags that are defined in the MetaLib display tables (refer to section B.4 below). You can use the DELETE option to prevent fields from displaying (it is possible that tags from the original format are the same as MARC21 tags or that an orignal MARC21 record has fields you want to suppress.)

Note that the system reads the table from top to bottom. This means, for example, that if you want to use a field to create a several fields and then delete the original field, the DELETE should be at the end.

Example:

```
In tag  In sub    Program                   Param       Action

331##   a                                   245,a
773#    x                                   022,a       ADDNEW
653##                                                   DELETE
650##   a                                               SPLIT
```

In the example above:

tag 331 subfield a is converted to 245 subfield a. The original 331 field is deleted
tag 773, subfield x is used to create 022 subfield a. The original 773 field is retained.
tag 653 is deleted
tag 650 is split into multiple fields. For example:

650 $$aSubject A $$aSubject B $$aSubject C

becomes:

650 $$aSubject A
650 $$aSubject B
650 $$aSubject C

Example for MERGE:

```
In tag  In sub    Program                  Param       Action

PL                                         260,a
PB                                         260,b       MERGE
YR                                         260,c       MERGE
```

In the example above the three fields PB, PL and YR will become one 260 field:

PL	New York
PB	Macmillan
YR	1999

become:

260 $$aNew York $$bMacmillan $$c1999

If MERGE had not been used, the following would have be created:

260 $$aNew York
260 $$bMacmillan
260 $c1999

Col. 6 Condition

The 'condition' column is used to limit the conversion of a field if another field is already present. The content of the column is:

NOTEXIST=code

For example:

```
In tag  In sub    Program       Param      Action      Condition
008          vir_fix_doc_first YR,a,8,4   ADDNEW
260       a                     YR,a       ADDNEW      NOTEXIST=YR
```

This means that the tag 260 is used to create the YR field only if the field has not already been created from the 008 tag.

3.2.8 Open URL

The 'Open URL' tab defines the routines used to create the openURL. The openURL is the means by which metadata from records is sent to the SFX server which then offer the user extended services.

The SFX sever requires very exact information. The original records, therefore, must undergo considerable parsing. The openURL can contain the following elements of information. The goal should be to send as much information as possible

- YEAR
- ISSN
- ISBN
- TITLE
- AUTHOR-FIRST - author's first name
- AUTHOR-LAST - authors's last name
- SUBJECT

The following elements are used only for articles:

- ARTICLE-TITLE
- VOLUME - volume number
- ISSUE - issue number
- START-PAGE

Note that the openURL is created from the *converted* record. There may be cases where fields are converted (in Conversion tab explained above) for the openURL rather than for display of the record in MetaLib.

Conversion program

In most cases the conversion program used is 'vir_950_standard' which means that the conversion table defined in below is to be used. If it is not possible to create a conversion table, a special conversion program can be created.

Conversion parameters

Parameters for a special conversion program.

3.2.9 Conversion Table

The following columns are defined:

Col.1 Field

The openURL element to be created. Select from the pull-down menu.

Col.2 Tag

The tag from the converted record to be used to create the openURL element.

Col. 3 Program

Special programs are used to process text if necessary (for the openURL it normally is necessary). The parameters for the program are entered in the 'Param' column At present the following programs are available. The programs are similar to the programs used by the 'Conversion' tab described above.

vir_950_first

Defines initial position and length of string to take from the beginning of the tag/subfield. Note that the count starts from position 1.

vir_950_last

Defines initial position and length of string to take from the end of the tag/subfield. Note that the count starts from position 0.

vir_950_parser

Parser program to extract text from the middle of tag/subfield.

vir_950_au_firlst

This program is used to parse the author (first/last) when the author's first name appears before his last name.

vir_950_au_lstfst

This program is used to parse the author (first/last) when the author's last name appears before his first name.

<u>Col. 4 Parameters</u>

This column is used for two purposes:

1. To defines the parameters used by the program defined in the third column.
2. If no program is used, the column is used to define the subfield/s used from the tag defined in the second column.

The following types of parameters may be used.

vir_950_first
This parameters for this program are:
'P' and subfield used from the tag defined in the second column. If the tag does not contain subfields, just enter 'P'.
'F' and the initial position and length of string.

```
Field Tag         Program            Param
YEAR  008   vir_950_first            P,F8,4
```

This means that 4 characters starting from the 8th position from tag 008 should be used to create the YEAR.

```
------>
         12345678
      008 000814s1975----dcu------b----00010-eng--
```

vir_950_last
The parameters for this program are:
'P' and subfield used from the tag defined in the second column. If the tag does not contain subfield, just enter 'P'.
'L' and the initial position and length of string.

```
Field Tag         Program            Param
YEAR  510##       vir_950_last       Pb,L3,4
```

This means that the last four characters from subfield b of tag 510 should be used to create the YEAR

```
                  3210
                  <---
      510   b  Jan, 1999
```

vir_950_parser
The parameters for this program are:

P - program parameter. Subfield to be used. This can be blank.
\S - indicates the start of parsing
\E - indicates the end of parsing. Use 'E*' to indicate that all text to the end of the line. E without an asterisk indicates that text to the first space or dash should be taken. Note that spaces do not have to be defined.

The parameters P, \S and \E must be used. Additional parameters:

\A - Used to indicate any alphabetic character. This parameter is used in conjunction with \S or \E to indicate the start or ending of parsing from the first alphabetic character.
\N - Used to indicate any numeric character. This parameter is used in conjunction with \S or \E to indicate the start or ending of parsing from the first numeric character.

For example:

In order to create the 'START-PAGE' field from the following:

773 t Journal of School Health |g v69 n9 p347-55 Nov 1999

```
Field        Tag         Program            Param
START-PAGE   773##       vir_950_parser     Pg\Sp\E-
```

This means that the text starting from 'p' and ending with '-' in tag 773, subfield g should be used to create the START-PAGE openURL element.

vir_950_au_firlst/ vir_950_au_lstfir
The parameters for these programs are 'P' and the subfield/s from which the author should be taken. For example:

```
Field              Tag         Program                  Param
AUTHOR-FIRST       100##       vir_950_au_lstfir        Pa
AUTHOR-LAST        100##       vir_950_au_lstfir        Pa
AUTHOR-FIRST       700##       vir_950_au_lstfir        Pa
AUTHOR-LAST        700##       vir_950_au_lstfir        Pa
```

This means that both the first and last names of the author will be taken from subfield a from the 100 tag. The vir_950_au_lstfir program is used because the author appears like this:

10010 |a Lowry, Richard

Note that if the 700 field is used it must be defined as well.

3.2.10 Applications

This tab defines various aspects of the resource that are used by applications using the Universal Gateway.

WEB-ADDRESS

This parameter is the URL of the home/main page of the resource used to enable direct access to the resource from the Resource List.

Example:

http://casanova.ub.hu-berlin.de:4505/ALEPH/-/start/hub01

WEB-ADDRESS-DIRECT

This parameter defines the URL that enables the user to "jump-to" the record in the target database from the record retrieved by MetaLib (that is, rather than a link to the home page, this link provides a link to the record itself). This functionality can be very important, as it enables the user to check holdings information available in the source. In order to be able to create this URL, the correct "link-to" syntax for the target database must be available. The ID of the record in the target database is entered by MetaLib in the $0100 placeholder. The ID of the source record is stored in the MetaLib SID field. The creation of the SID field is controlled by the IN-DIRECT-KEY and IN-DIRECT-KEY-PARAM parameters explained above.

The following example is the direct "link-to" syntax for a sample ALEPH500 database.

Example:

http://casanova.ub.hu-berlin.de:4505/ALEPH/-/start-ext/hub01?direct-doc/hub01-$0100-999-sysno-nocont

→ Note: you cannot enter an ampersand ('&') in an HTML form. Enter a double plus sign ('++') in its place; the system stores and sends an ampersand.

WEB-ADDRESS-ITEM

This parameter defines the URL that enables the user to "jump-to" the items (holdings) of the record in the target database from the record retrieved by MetaLib. This URL is used when the HOLDINGS-METHOD is 'JUMP'. In order to be able to create this URL, the correct "link-to" syntax for the target database must be available. The ID of the record in the target database is entered by MetaLib in the $0100 placeholder. The ID of the source record is stored in the MetaLib SID field. The creation of the SID field is controlled by the IN-DIRECT-KEY and IN-DIRECT-KEY-PARAM parameters explained above.

METALIB-INFO (Y/N)
Not in use.

METALIB-SELECT (Y/N)
Not in use.

METALIB-CONNECT (Y/N)
Not in use.

METALIB-NOTE
Not in use.

Z39-SERVER-REC-TYPE & Z39-SERVER-REC-FIX & Z39-SERVER-CHAR-CONV
These parameters are required when MetaLib's Universal Gateway locates records for MetaLib's z39_server. This z39_server is used, for example, by SFX in order to fetch additional information about the records (for example: to fetch additional authors; the openURL contains only the first author). It is also used by for a distributed search for cataloging which is an optional add-on to MetaLib. These parameters define the conversion that should be applied to the records retrieved by the Universal Gateway before they are sent back to the requester.

Z39-SERVER-REC-TYPE
Defines record format.

Z39-SERVER-REC-FIX
Defines a fix program. The Universal gateway will send the records in their original format. If the z39 client requires a different format, a conversion program will be needed.

Z39-SERVER-CHAR-CONV
Defines character conversion program. All records are in UTF-8.

Since the Z39 server may receive requests for different formats, up to 5 conversions may be defined.

Example:

```
Z39-SERVER-REC-TYPE-1   USMARC
Z39-SERVER-REC-FIX-2    vir_fix_pubmed
Z39-SERVER-CHAR-CONV  UTF_TO_8859_1
```

LOCATE
In development.

HOLDING-METHOD
The can be used when it is possible to create a direct link to the items or holdings information in the MetaLib target. When this field is filled in, MetaLib will display a “Holdings” field in the full record. This field is underlined and can be activated to display the holdings of the record. There are two methods:

1. OPAC indicates that the target supports the Z39 OPAC syntax for the retrieval of holdings information. MetaLib sends a Z39 request to the target and displays the results. OPAC also works with the ALEPH500 protocol (version 12.1, 12.2 and 15.1).

2. JUMP indicates that a WEB-ADDRESS-ITEM field has been defined for the resource. When the holding field is clicked on, MetaLib activates the link to the targets’ holdings screen for the specific MetaLib record.

3.3 Z39_Gate Configuration

If the MetaLib resource is accessed using Z39, it must have a configuration file for the Z39 gateway server in the $alephe_tab/z39_gate directory.

Note that the ‘target’ defined in the Z39 configuration file must be the same as the name of the MetaLib resource code (RESOURCE).

The ‘database’ in the Z39 configuration must be the same as the DATABASE field defined in the MetaLib resource configuration.

For example:

If the MetaLib DATABASE code is ERIC_SOCIO and the BASE is SOCIO, the Z39 configuration file should be defined as follows:

```
target        EBSCO_SOCIO
database      SOCIO
```

Refer to the attached documents “Z39_gate” and “Z39_gate_client” for more information on configuring and testing the Z39 gateway server.

4 MetaLib Access Management – Authentication & Authorization

Users can access MetaLib as "Guest" users or with individual logins which enable them to create their own user profile and enjoy special privileges and services, such as e-shelf. Users can either be self-registered or the system administrator can choose to pre-load user records into MetaLib or use the relevant institutional authentication server. For details on the latter, refer to Ex Libris MetaLib support.

MetaLib users are stored in an Oracle table, Z312. The MetaLib user record includes the following details:

- ID
- Name
- Password
- Title
- Address/Zip/Email/Telephone
- Academic status (note field)

In addition to the above, there are two important fields:

1. **Institution**

Several institutions may use the same MetaLib installation. The 'institution' field can be used to affiliate a user with a certain institution or part of the institution for authentication and/or authorization purposes, as explained below.

When logging into MetaLib, a user is requested to select the institution to which he is affiliated. If there is only one institution, this selection field can be hidden in the HTML screen (login-1).

Guest users and self-registered users are automatically assigned the institution "metalib" . This field cannot be updated in the self-registration or user update form.

Each institution should have a defined user with a user ID that is the same as the institution code. For example, the institution 'metalib' has a user with the ID 'metalib'(default password, 'metalib'). The resource list defined for the institution's user serves as the default for new users belonging to the same institution. Every MetaLib installation should create a user 'metalib' with a default resource list for guest users. Self-registed users can modify this list as well as personalize other aspects (create alerts and save records to the e-shelf).

The 'institution' field can have up to 100 characters.

2. **Status**

The 'status' field can be used to distinguish between different types of users within one institution for authorization purposes, as explained below. Guest users and self-registered users are automatically assigned a default status of 'guest'. The field cannot be updated in the self-registration or user update form.

The ‘status’ field can have up to 10 characters.

4.1 User Authentication

A user must log in to be able to work with MetaLib. The user can access MetaLib as a “guest” user or enter his own ID and password.

A user gaining access as a “guest” is not authenticated. His authorizations will be those defined for the “guest” user, as specified in the ‘Metalib’ user record and in the authorization table.

A user gaining access with an individual login can be authenticated either locally (i.e., against the MetaLib users file) or against an external authentication server. The authentication method is specified in a table called tab_authentication in the tab directory of vir00.

Structure of the table:

Col. 1 Institution code. This is the institution that is defined in the MetaLib user record and is entered in the MetaLib login screen.

Col. 2 Authentication type.
‘local’ – indicates that the user is authenticated against MetaLib’s users file.
program name – name of program used for authentication via a remote authentication server.

Col. 3 Host
Host IP and port for accessing the external authentication server.

Example:

```
!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!-!!!!!!!|!|!|!
metalib             local
exlibris-usa        aleph_14_2          ram42:6505
```

For more information on external authentication of users, refer to Ex Libris MetaLib Support.

4.2 User Authorizations

MetaLib contains a table which allows the installation to control access to MetaLib resources based on user ID, IP range, user institution and user status. You can define authorizations for each indvidual resource or associate a resource with a group and then define authorizations at the resource group level. The link between the resource and the group is created in the IRD (information resource database) record in the ‘Authorization Group’ field. Examples of groups are “Free” or “Restricted”. In a institution with campuses that subscribe to different resources, the group can be the campus. Note that every resource can belong to more than one group.

The table that defines authorizations is tab_authorize_source in the tab directory of vir00.

Structure of the table:

Note that the table is read by MetaLib from top to bottom. Asterisks may be used as wildcards in all columns.

Col. 1 User ID or an asterisk to indicate all other users not previously defined.

Col. 2 &
Col. 3 IP OR beginning of IP range and end of IP range.

Col. 4 User institution or an asterisk to indicate all institutions not previously defined

Col. 5 User status or an asterisk to indicate all statuses not previously defined

Col. 6 MetaLib Resource ID or an asterisk to indicate all resources not previously defined. If you intend to use the resource authorization group, enter an asterisk for all lines.

Col. 7 Authorization Group or an asterisk to indicate all groups not previously defined.

Col 8   Action (Y/N).
Y = user is allowed to access resource
N = user is not allowed to access resource

Example:
The following example is for a University with two different campuses, each with its own set of resources. A campus can be defined as a specific IP range or as different instiutions (as long as users from each campus are assigned the different institutions). In the example below, each campus is a separate MetaLib institution.

In the IRD databases:

resources that "Campus A" subscribes to are assigned to the 'Authoization group'=CAMPUS-A

resources that "Campus B" subscribes to are assigned to the 'Authoization group'=CAMPUS-B

resources that both "Campus A" and "Campus B" subscribe to are assigned to the 'Authoization group'=CAMPUS-A & CAMPUS-B.

free resources (that is, those that do not require a subscription) are assigned to the 'Authorization group'=FREE

In the user record:

Users that belong to "Campus A" have 'Institution'=CAMPUS-A

Users that belong to “Campus B” have ‘Institution’=CAMPUS-B

Guest users have the ‘Institution’=METALIB

Note that the length of the columns is not accurate in the example below.

```
   1      2       3        4         5        6          7          8
!!!!!!!!-!!!!!..-!!!!!..-!!!!!!!!..-!!!!!..-!!!!!!!..-!!!!!!!!!..-!
*        *       *       CAMPUS-A   *       *         CAMPUS-A    Y
*        *       *       CAMPUS-B   *       *         CAMPUS-B    Y
*        *       *       *          *       *         CAMPUS-A    N
*        *       *       *          *       *         CAMPUS-B    N
*        *       *       *          *       *         FREE        Y
```

Explanation

All users that belong to the MetaLib institution ‘CAMPUS-A’ (regardless of IP or user status) are allowed to use resources belonging to authorization group ‘CAMPUS-A’ or ‘FREE’

All users that belong to the MetaLib institution ‘CAMPUS-B’ (regardless of IP or user status) are allowed to use resources belonging to authorization group ‘CAMPUS-B’ or ‘FREE’.

All other users (GUEST and self-registered users who will be automatically assigned to the institution ‘MetaLib’) will be allowed to use only resources belonging to the authorization group “FREE”.

Note:
If you want to define the user ID ‘guest’ define it as ‘guest*’ because MetaLib adds a running number to every user who logs in as ‘guest’.

For example:

```
guest*           *                    *      *                    Y
```

4.3 Bypassing the Sign-In Page

In principle, users must sign in to MetaLib to begin work. You can also bypass the login page and have users sign in at any stage during their work.

You can define the username, password and institution in the URL and in this way bypass the login screen. Use the following parameters:

F01 = username
F02 = password
F03 = institution

Normally users logged in this way are GUEST users.

You build the URL as follows:

```
http://<IP or URL of MetaLib server>:<port>/V/-/
        login?F01=<username>&F02=<password>&F03=<institution>
```

For example:

```
http://www.metalib.com/V/-/login?F01=GUEST&F02=GUEST&F03=metalib
```

5 MetaLib User Interface Setup

5.1 Helping the User to Locate Resources

Users can access the MetaLib search screen in three basic ways:

1. Via the **Information Gateway** -- by selecting one of the predefined groups of resources. The groups can be defined by the institution based on the Information Resources Database (IRD), based on subject, location or any other information in the IRD.
2. Via **Locate Resources** – by running a query on the Information Resources Database (IRD) and clicking on "GO".
3. Via **My Resource List** – by creating a personal list of resources based on a query (find) on the IRD. From the Locate Resources screen, click on "ADD" to add resources to the personal list. If the user already has a personal list, the system displays the MetaLib search screen with the user's personal resources.

The following sections explain how the predefined groups of resources and the Locate resources are set up.

5.1.1 Info Gateway Pre-defined Groups of Resources

On the Info Gateway screen, the institution can define predefined groups of resources for the user to select from. These groups are created by defining a FIND command in the HTML file. The FIND command is based on the 'WFL' word index that is created from the Info Gateway Group field (IRD field code: FIL).

The name of the file is 'source-pre' in www_v_lng:

```
>> $alephe_root/www_v_lng    OR   wv
>> vi source-pre
```

For example:

```
<a href="&server_vir/source-pre-search?F-WFL=Agriculture&F-WST=act
ive&GROUP_TEXT=Agriculture resources" target="error">Agriculture</a>
<br><a href="&server_vir/source-pre-search?F-WFL=Anthropology&F-
WST=ac
tive&GROUP_TEXT=Anthropologyi resources"
target="error">Anthropology</a>
<br><a href="&server_vir/source-pre-search?F-WFL=Area Studies&F-
WST=ac
tive&GROUP_TEXT=Area Studies resources" target="error">Area
Studies</a><br>
```

Note that the FIND request always includes a search on the WST (status word file) to request only 'active' IRD records

You can also define a FIND command with the 'OR' boolean operator. For example:

```
<a href="&server_vir/source-pre-search?F-TGT=JMSJMS OR F-WFL=Agric
ulture&F-WST=active&GROUP_TEXT=Agriculture                resources"
target="error">Agriculture
</a><br>
```

This can be useful in order to always add the library's own OPAC to all Info Gateway groupings.

5.1.2 Locate Resources

The Locate Resources function searches IRD indexes. The indexes searched are defined in the HTML file – 'source-find-include'. List of values to choose from can be defined (for example 'dod-collection' for the list of resource types).

Note that the FIND request always includes a search on the WST (status word file) to request only 'active' IRD records

5.1.3 Resource List Names (GROUP_TEXT)

When the user accesses the MetaLib search screen, the system displays, above the list of selected resources, a message such as "Search **My resources**" or "Search **Mathematics resources**". This message is combined of two elements:

1. The name of the group of resources the user selected. This name is defined as part of the HTML link as the "GROUP_TEXT" (for example: '**My resources'** or "**'Mathematics resources'**
2. Text that is defined in the HTML file ('Search')

GROUP_TEXT names are defined in the following HTML files. Spaces in the text should be indicated by '%20'.

1. source-pre

A GROUP_TEXT is defined in this file for each predefined grouping and for the link to the 'My Resource List'

For example:

```
<a href="&server_vir/source-pre-private?GROUP_TEXT=My Resources" ta
rget=error>
<a href="&server_vir/source-pre-search?F-WFL=(Arts and Architectur
e)&F-WST=active&GROUP_TEXT=Arts%20and%20Architecture%20resources    "
target="error">Arts
 &#38; architecture</a>
```

2. source-privated

A GROUP_TEXT is defined in this file for the resources selected by the user after running a 'Locate Resource' find.

Example:

```
window.document.form1.GROUP_TEXT.value = "Selected%20resources";
```

3. source-private-list-head

A GROUP_TEXT is defined in this file for the link to “GO” to the personal resource list from “Locate Resources”

Example:

```
My Resource List  <a href="&server_vir/source-pre-
private?GROUP_TEXT=My%20resources" target=error><img
src="&icon_path_&lng/icon/v-go-icon.gif" border="0" alt="Search My
Resources"></a>
```

4. source-private-pre-0

A GROUP_TEXT is defined in this file for the situation where the user logs in directly to the MetaLib search screen because he has a personal resource list.

Example:

```
<body onload="top.location = '&server_vir/source-pre-
private?GROUP_TEXT=My%20Resources'">
```

The text is entered in the various MetaLib search screen HTML files. There are several such files:

1. find-head

Enter here the text for the list of resources that can be searched via MetaLib search. For example:

```
Search <span class="text2" id="normalb">$1200</span>:
```

The ‘$1200’ is the place holder for the GROUP_TEXT defined above,

2. find-head-1

Enter here the text for the list of resources that can only be linked to directly. For example:

```
<tr class=tr1>
    <td class=text3 colspan=5 nowrap align=left>
      Link to additional <span class="text2"
id="normalb">$0100</span>:
    </td>
```

The ‘$0100’ is the place holder for the GROUP_TEXT defined above,

3. find-ext-head

Enter here the text for the list of resources that can only be linked to directly and when the user chooses only resources that can be linked to directly. For example:

```
<table cellspacing=1 border=0 width="100%" empty-cells="show">
<tr class=tr1>
   <td class=text3 colspan=5 nowrap align=left>
    Click on one of the $0100 below to link to it directly
   </td>
```

In addition to these files the GROUP_TEXT for the resources that are displayed after a saved query is re-run is defined in a message (number 163) in the www_v_heading messages file:

```
0163 0000 L Saved search resources.
```

To edit the file:

```
>> cd $alephe_root
>> cd error_eng
>> vi www_v_heading
```

5.1.4 Resource Search Limit & Automatic Selection of Resources

Resources that can be searched via MetaLib search (that is, they have a Z58 resource configuration record) can be searched together. MetaLib limits the number of resources that can be searched together. This limit can be defined by the institution in www_server_defaults:

```
setenv www_metalib_search_limit       08
```

To edit the file:

```
>> cd $alephe_root
>> vi www_server_defaults
```

When a user selects a pre-defined group of resources from the Info Gateway screen or selects records after doing a ‘Locate Resources’ and then clicks on “GO”, MetaLib presents the user with the MetaLib search screen and automatically selects the resources if they are less then the defined limit.

5.2 Merging and De-Duplication of Sets

MetaLib can merge and de-duplicate sets. The number of records that can be included in a merged set can be set by the institution. Note that the merge process can take some time mainly due to the need of the system to fetch the records from the targets.

The number of records that can be merged is defined in www_server_defaults in the following parameter:

```
setenv www_metalib_merge_limit       150
```

To edit the file:

```
>> cd $alephe_root
>> vi www_server_defaults
```

Note that MetaLib requires a resource configuration record for the merged set:

> MERGESET - the RESOURCE-SHORTNAME, 'Merged Set' is used as the header for merged set.

The resource requires only the RESOURCE and RESOURCE-SHORTNAME fields.

5.2.1 De-Duplication Algorithm

The algorithm used for de-duplication is defined in the union_match_kobv_tag table in the tab directory of the VIR00 library:

```
>> dlib vir00
>> dt
>> vi union_match_kobv_tag
```

Structure of the table:

Col. 1 Field code:
Field code to be used for match.
Col. 2 Not in use
Col. 3 Match
Positive weight for match. Records match if sum of all positive weights of the matching fields is larger than the threshold.
Col.4 Non-match
Negative weight for match. There is no match if the sum of all negative weights of the non-matching fields is larger than the threshold. Non-match is stronger than match.
Col.5 Normalization
Normalization routine identifier. Currently this must be "1" which normalizes in the same way as the word index (for example, remove special characters, capitalize/uncapitalize characters, etc).
Col.6 Not in use
Col.7 Compare routine
Values are 'a' or '3'. This defines the compare routine for detecting a precise match.

- a = exact comparison in consideration of other parameters (normalize, length, delimiter, etc.).
- 3 = tree-gram comparison. This is a special kind of comparison to check the similarity of fields (title, author, and so on).

Col.8 Sub-field code
Sub-field(s) to be used.

Col.9 Delimiter
When taking text for comparison, use only the text up to this delimiter.
Col.10 Length
Length of the text to be compared.
Col.11 Characters
Characters to be used when performing the comparison.

Example:

```
! 1    2   3   4   5 6 7        8              9    10    11
!!!!!-!!!-!!!-!!!-!-!-!-!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!-!!!!!!!!>

020## 075 070 010 1 0 a a                             0123456789x
022## 075 070 010 1 0 a a                             0123456789x
100##       100       030       100       1       0       3       a
abcdefghijklmnopqrstzuvw
xyz0123456789
110##       100       030       100       1       0       3       abcd
abcdefghijklmnopqrstzuvwx
yz0123456789
111##       100       030       100       1       0       3       acde
abcdefghijklmnopqrstzuvwx
yz0123456789
130##       100       030       100       1       0       3       adlnop
abcdefghijklmnopqrstzuvwx
yz0123456789
245##        100        070        100        1        0        3
abcdefghijklmnopqrstzuvwx
yz0123456789
260## 000 070 030 1 0 a c                             0123456789
YR### 000 070 030 1 0 a c                             0123456789
910##       000       070       000       1       1       a       a
abcdefghijklmnopqrstzuvwx
yz0123456789
```

The thresholds to determine whether or not the record is a duplicate are defined in the table ‘union_match_kobv_param’ which is also in the tab directory of VIR00.

5.3 Contact

One of the options in the MetaLib banner is “Contact” which enables the user to send an e-mail message to the institution. The e-mail address is defined in the HTML file ‘source-include-map’:

```
>> wv  OR  $alephe_root/www_v_eng
>> vi source-include-map
```

For example:

```
<area shape="rect"
      coords="583,22,647,38"
```

```
    href="mailto:nina@exlibris.co.il"
    target=_top
    alt="Contact us">
```

6 MetaLib System Maintenance

MetaLib has some server based utilities for helping to maintain the system. To access these utilities, move to the environment of one of the MetaLib libraries and enter the command : 'UTIL':

```
>> dlib vir00
>> util
```

A menu displays. Only some of the options are used in MetaLib and are mentioned below.

6.1 MetaLib Background Processes

Several processes must be running for MetaLib to work:

Z39 gateway (z39_gate)
Z39 server (z39_server)
PC server (pc_server)
Apache/WWW server (www_server)
Oracle

By convention, the following ports should be used:

Process	version m505	version m515
z39_gate	9907	9917
z39_server	9909	9919
pc_server	6505	6515
www_server	4501-	4511-

6.1.1 Monitoring MetaLib Processes

The Z39 gateway, Z39 sever, PC server and WWW servers can be monitored by using the 'server_monitor' command.

```
>> server_monitor
```

The following processes should be displayed:

```
| Port  | Pid   | Server Type  | Started At      | Status
|---------|---------|-----------------|-----------------|--------------------
| 4501  | 1565  | WWW Server   | Sep 15 20:56:23 | Free
| 4502  | 1566  | WWW Server   | Sep 15 20:56:23 | Free
| 4503  | 1567  | WWW Server   | Sep 15 20:56:23 | Free
| 4504  | 1568  | WWW Server   | Sep 15 20:56:23 | Free
| 6505  | 20775 | PC Server    | Sep 10 10:52:28 | Free
| 9907  | 24110 | Z39 Gate     | Sep 14 15:34:30 | Free
| 9909  | 20791 | Z39 Server   | Sep 10 10:52:38 | Free
```

Apache can be monitored with the ‘ps’ command:

```
>> ps –ef | grep apache
```

On Linux machines use:

```
>> ps –efw | grep http
```

Oracle can be monitored by using UTIL O.

6.1.2 Starting MetaLib Processes

All MetaLib processes are defined to start automatically when the computer is booted. If necessary, these processes can also be started manually, as follows:

6.1.3 Apache, WWW servers, z39_gate, Z39_server

These servers can be started using UTIL W-3. They can also be started manually:

To run the pc_server:

```
>> pc_server 6505 &   OR     pc_server 6515
```

To run z39_gate:

```
>> z39_gate 9907 &   OR      z39_gate 9917 &
```

To run Z39_server:

```
>> z39_server 9909 & OR     z39_server 9919 &
```

To run apache:

```
>> su root
for m505:

>>  /aleph/product/apache_1.3.12/bin/httpd –d /aleph/a50_5/apache

for m515:

>>  /aleph/product/apache_1.3.12/bin/httpd –d /aleph/a51_5/apache
```

Apache automatically starts the WWW servers.

6.1.4 Configuring Additional WWW Servers

MetaLib's WWW servers are started by Apache. The number of servers that are started and their port numbers are defined in $alephe_tab/www_front_cgi.conf.:

```
>> cd $alephe_tab
>> vi www_front_cgi.conf
```

Example:

```
! www_server port number
!-------------------------------------
4501
4502
4503
4504
```

Add additional port numbers to increase the number of servers.

6.1.5 Logfiles

Logfiles of the www_servers are in the directory $LOGDIR:

```
>> cd $LOGDIR
```

The name of the file is www_server_<port>.log

Logfiles of the Z39 gate are in the directory $TMPDIR

```
>> cd $TMPDIR
```

The name of the file is z39_gate_<port>.log

6.2 MetaLib Backup

Data from the VIR00 and DAT01 libraries needs to be backed up. The data in these libraries include:

- Resource cataloging & configuration record (DAT01 and VIR00)
- User records: users, profiles, alerts, history

The frequency of the backup should depends on the amount of updating in these files. If work on resource configuration is minimal, a backup should be run at least once a week. If there is extensive work on resources the backup should be more frequent.

Before backup the data from these libraries should be exported. This is done with the following commands:

```
>> dlib dat01
>> ap
>> exp_ora_library

>> dlib vir00
>> ap
>> exp_ora_library
```

The export files are created in the 'files' directory under each library:

```
>> dlib dat01
>> cd files  OR df1

>> dlib dat01
>> cd files  OR df1
```

In addition to the data in the two libraries, the MetaLib system should be backed up. This is done by backing up the directory tree under …/a50_5 (or /a51_5 for version m515). The backup includes the export files from the dat01 and vir00 libraries.

6.3 MetaLib Maintenance Procedures

MetaLib stores retrieved records in the vir01 library; these records are not needed peramanently and need to be deleted periodically. In addition, MetaLib creates scratch and log files that should be deleted. The following "clean-up" procedure should be run at least once every three days:

```
>> ap
>> clear_vir01
```

This procedure can be added to a script and run on an automatic basis. If you require help in setting this up, please refer to MetaLib support.

The procedure can also be run from the /M management interface - "Clear Temporary Storage".