Alephino Configuration

From: Swami Wiki **Date: August 15th 2012**

Table of contents

General Information

The central configuration file of the Alephino server **alephino.cfg** is located in directory **etc**. Apart from parameters for controlling the server all references to files used by the server have their origin here. The file is organized sectionwise. A section starts with a name surrounded by round brackets. On occurrence of equal sections their contents will be combined. Each parameter line has the format **Parameter name = Parameter value**. The names of sections and parameters are not case sensitive, upper- and lowercase writing will be dealt with equally.

A parameter line may cover up to 200 characters. The equal sign between name and value may be enclosed by any number of spaces. Tab characters are not allowed.

Lines beginning with an asterisk * will be dealt with as comments.

Parameter section (Communication)

This section contains global parameters for Client/Server communication.

Example:

```
(Communication)
Host            = localhost
Info            = ../etc/pc_service.dat
Port            = 2069
MaxConns        = 20
Compress        = 3
VersionId       = 18.01
VersionId       = 4.00
Mail            = smtp.t-online.de
*Logfile         = ../temp/alephino.log
*MailProt        = ../temp/smtp.log
```

Explanation:

Host= The server configuation file serves also for controlling the Web service module. This parameter contains information about the Alephino server to be used by the Web services.

Info= Path name of the file containing short descriptions of every service of the Alephino server. (These descriptions are written to standard output while operation of the server.)

Port= The TCP port number the Alephino server is listening to.

MaxConns= The maximum number of server connections managed simultaneously.
Default: 50

Compress= The logical number of a compression method. The data transfer between Alephino server and applications (clients) happens in a compressed form. By default compression level **3** has to be used.

VersionId= The number of the C/S protocol specification supported by the Alephino server. This declaration is repeatable. With that the server can be directed to accept multiple version numbers as long as they're indicating the same protocol specification. This is necessary because Aleph and Alephino do use different version numbering.

NonBlock= Several functions of the Alephino server do need TCP/IP client activity. This parameter is to define the maximum time (in seconds) the Alephino server acting as client waits for an outbound connection to be established. In the absence of this parameter the server might be blocked for an undefined period of time. The parameter can be overridden if also located in additional specialized parameter sections.

Wait= If the server is acting as TCP/IP client, this parameter defines the maximum wait time (in seconds) for the completion of a transaction. **Default: 10 seconds**

MaxIdle= This parameter defines the time span where the server is waiting for an existing connection to be reused resp. triggered. After this time (in seconds) the server closes the connection so the client has to establish a new one on next access.
Default: 300 seconds

Mail= If the server shall be used for sending email (Reminder letters, orders etc. via Web service module), the network name or IP address of the SMTP server has to be declared here.

Logfile= For debugging purposes all in- and outgoing Alephino server messages can be recorded. For this a file name has to be added here. Pls. note that this file tends to grow rapidly!

MailProt= For debugging of the email transmission process all SMTP messages can be recorded. For this a file name has to be added here. Also note that the protocol file may grow rapidly, so switch back to normal mode asap.

Parameter section (Shutdown)

The Alephino server has the ability to shutdown itself at given time.

Example:

```
(Shutdown)
Time       = 23:45
Weekday    = Mon-Fri
```

Explanation:

Time= Time to shut down written HH:MI.

Weekday= The automatic shut down can be limited to certain weekdays. Pls. use the 3 digit code of the respective weekday. A range of weekdays can be given using **-** as separator.

Parameter section (Sublibs)

This section covers parameters for controlling the sub-library system.

Example:

```
(Sublibs)
SublibMode = N
Default = ZB
```

Explanation:

SublibMode=Y/N With this parameter set (**Y**), certain proceedings in circulation, in particular the hold requesting and delivery, will be controlled depending on the sub-library the involved borrower and items belongs to.

Default= The code for the sub-library default. This code will always be used as long as the data does not contain a sub-library.

Parameter section (Files)

This section covers references to files that make the set of control tables of the Alephino database and server.

Example:

```
(Files)
License  = ../etc/alephino.lic
UserTab  = ../etc/alephino.usr
```

```
DataDict = ../etc/marcdata.int
DataDict = ../etc/marcindex.int
DataDict = ../etc/marclink.int
DataDict = ../etc/marcdoubl.int
DataDict = ../etc/marcnumb.int
DataDict = ../etc/marctransl.int
ApplDict = ../etc/marcfile.ext
ApplDict = ../etc/marcindex.ext
ApplDict = ../etc/marclink.ext
ApplDict = ../etc/marcisbd.ext
ApplDict = ../etc/marcform.ext
ApplDict = ../etc/marclist.ext
ApplDict = ../etc/marcsort.ext
ApplDict = ../etc/marcscript.ext
ApplDict = ../etc/marclocate.ext
ApplDict = ../etc/marctable.ext
ApplDict = ../etc/marcstat.ext
ApplDict = ../etc/addresses.tab
ApplDict = ../etc/printcat.ext
ApplDict = ../etc/printmarc.ext
ApplDict = ../etc/printcir.ext
ApplDict = ../etc/printcsh.ext
ApplDict = ../etc/printacq.ext
ApplDict = ../etc/printsta.ext
ApplDict = ../etc/printivh.ext
ApplDict = ../etc/sipform.ext
ApplDict = ../etc/printris.ext
SetFile = ../temp/alephino.set
CatTable = ../etc/client/pc_%s.pck
ServerLog = ../data/aserver.log
LicenseLog = ../data/license.log
UtilLog = ../data/adm_util.log
UtilProt = ../data/adm_util.prt
JobLog = ../data/job.log
JobSeq = ../data/job.seq
TempIndex = ../temp/index.dat
Statistics = ../etc/statistics.cfg
XStyle = ../etc/xslt/%s.xsl
XDebug = ../data/xdebug.xml
XInput = ../data/xinput.xml
HTTPDebug = ../data/htdebug.txt
```

Explanation:

License= Path name of the Alephino license file

UserTab= Path name of a file containing system wide staff user accounts. In the original state staff users **SYS** and **WWW** are pre-defined.

DataDict= Path name of an database control table of the "internal" type. This parameter is repeatable since the internal control structure is usually held in multiple file.

ApplDict= Path name of an database control table of the "external" type. This parameter is repeatable since the external control structure is usually held in multiple file.

SetFile= Path name for temporary files used to manage results sets while operation of the server.

CatTable= Path name for so GUI client packages. These are files, located at the server, that will be transferred to the GUI client and unfold in local client directory on client start. CatTable does contain a placeholder **%s** that is automatically replaced by the 3 digit code of the currently opened Alephino master file. The following rule applies:

If a package file does exist, whose name refers to a specific data pool it has precedence over a generic version of the same package.
Example: There are 2 files ../etc/client/pc_TIT.pck and ../etc/client/pc_X-TIT.pck.
On client package request for master file „TIT(les)“ of the data pool „X“ the server sends the special package pc_X-TIT.pck, otherwise (for all other data pools) the generic package pc_TIT.pck.

ServerLog= Path name of a file where all start and stop actions of the Alephino server are recorded. Will be used for error analysis.

LicenseLog= Path name of a file where license limit overflows will be registered. Can be used for indication of license increments.

UtilLog= Path name for the log file of Alephino server administration. All relevant administrative actions performed since the time of installation regardless whether via Web service module or standalone batch utility **adm_batch** are recorded here.

UtilProt= Path name pattern for protocol files of the Alephino server administration. Each relevant administrative action is recorded with its parameters and detailled results in a separate file. The file name is build of this path name pattern followed by the automatically created job number (4-digits), separated by a dot. Protocol file names will be reused resp. overwritten after the pre-defined log cycle.

JobLog= Path name for log book. By default the most recent 100 administrative transactions are recorded. The list can be browsed anytime using Web services "Administration->Show log".

JobSeq= Path name for managing the administrative log (see. JobLog)

TempIndex= Path name of temporary file used for browsing the index (scan lists).

Statistics= Configuration file containing declarations for pre-configured statistical reports that can be used out of the Web service module.

XStyle= Path name pattern for XSL style sheets. Style sheets are used to control the process of formatting printouts and emails based on XML data extracts. The placeholder **%s** is replaced by the name of the respective style sheet which is given by **STYLE=** declarations from external server control tables.

XDebug= (Option) Temporary file that contains a recording (XML formatted) of the most recently performed XSL transformation. This file is helpful for debugging purposes in combination with the creation of printouts or email.

XInput= (Option) Temporary file containing recorded input data (XML formatted) of the X-Services used for online data exchange and Web based data editing. The file is needed for debugging purposes.

HTTPDebug= (Option) Temporary file for recording of the most recent HTTP dialog inititated by the Alephino server. This file can be applied for debugging purposes.

MessBalance= Path name of a file that is used to manage update messages received from an Aleph consortium server. Based on this messages the replication process of bibliographic data to the Alephino local system is controlled.

MessUpload= Path name of a file that is used to manage update messages sent to an Aleph consortium server. It covers updates to items and local holding records. Based on this messages the replication process of local holdings information from local Alephino system to an Aleph based consortium is controlled.

Parameter section (Limits)

This section covers definitions of various Alephino server limits.

Example:

```
(Limits)
MaxPrintList = 100
MaxJobNo = 100
LoanHistory = 3
```

Explanation:

MaxPrintList= Maximum number of records for printout. A reasonable limitation should be set anyway. Otherwise OPAC users have the opportunity to slow down the entire system by extensive use of printout creation if applied to large result list.

MaxJobNo= Maximum number of recorded administrative transactions (log book cycle). By default the most recent **100** transactions are available via "Show log".

LoanHistory= Maximum number of recorded history loans per item. **Default: 1**
On overflow the oldest data record will be overwritten (first-in-first-out).

MaxRecLen= Maximum length of a data record. **Default: 20.000**
Pls. note that the GUI can't deal with data records longer than approx. 45.000 characters.

MaxFldLen= Maximum length of a data field. **Default: 7.500**
Pls. note that the GUI does cut field contents from 2.000 characters.

MaxTrmLen= Maximum length of a sort term. **Default: 256**

MaxQryLen= Maximum length of a search request. **Default: 500**

MaxRecords= (Currently not in use.)

MaxFields= (Currently not in use.)

MaxResults= Maximum number entries / page for scan lists. **Default: 10.000**

MaxSets= Maximum number result sets that can be handled simultaneously. **Default: 5.000**

MaxRetTime= Maximum execution time for retrieval and sorting (in seconds). **Default: 120**

IgnoreTimeout=Y/N With Y the above time-out while retrieval can be ignored.**Default: N** (Necessary on import of large numbers of data where the time-out on internal retrieval actions while linking records must not lead to abort.)

MaxScanCnt= Maximum number of linked records displayed for each index entry on scan list creation. **Default: 100.000**

MaxTerms= (Currently not in use.)

MaxAspects= Maximum number of search aspects to consider while retrieval and filtering. **Default: 100**

MaxEdtStk= Maximum stack depths (number recursions) on display or print formatting controlled by sub-formats (table driven record formatting). **Default: 10**

IdxBufLen= Buffer length for index word creation from field contents. **Default: 10.000**
On creation of word indexes the buffer must cover the respective term length (according to TERM definition) multiplied with the number of words to be indexed. Given the term is declared 100, 100 words at maximum from the respective field contents will be considered for index. If the field does contain more than 100 words (common for abstracts) this number must be increased.

RetBufLen= Buffer size for database access on retrieval. **Default: 2048**

Parameter section (Templates)

Example:

```
(Templates)
Date = MM/DD/YYYY
```

Explanation:

Date= Pattern for common date formatting.

ISOSet= Path name of a file containing a character translation table. Background: Although the Alephino always uses Unicode (UTF-8) internally, all text files at the server have character set ISO8859-1 (Latin-1). The cause is that files like this can be dealt with by means of the system editor tool on any supported operating system. If special characters have to be entered that are not included in western european character set (Latin-1), an alternative character translation table can be applied here. For cyrillic texts a translation table covering ISO8859-5 to UTF-8 relations can be applied here.

MabIdent=Y/N This is only relevant in combination with the bibliographic standard MAB (used in Germany and Austria). According to the standard some field do have a prefix of fixed length (identication numbers) where the gaps are filled with spaces. Because the GUI removes multiple spaces by default, a temporary replacement must happen to allow interactive data import. With the parameter set to **Y** the server performs replacement of spaces by underline characters for the first 20 characters.

VersionTime=Y/N With the "Version-Check" only those server files will show up in the list that are newer than their counterparts at the client side. With that parameter set to **N** you can force all files to be listed. This is always useful if newly installed GUI clients shall be updated from server where special configuration files have been prepared.

ObjectAlias= **Base-URL** for addressing digital objects in Alephino. This parameter is used by the server while conversion of local object path names to fully qualified URLs.

Parameter section (Directories)

This section contains declarations of directories used by the Alephino server.

Example:

```
(Directories)
Temp = ../temp
Print = ../print
Scratch = ../temp
Backup = ../backup
ClientVersion = ../version
BorPict = ../data/photo
```

```
Upload = ../temp
Objects = ../data/objects
```

Explanation:

Temp= Directory for temporary files.

Print= Directory for ready-to-print (or email) files.

Scratch= Directory for printout files that have already be processed by the print deamon.

Backup= Directory for backup copies of files processed via Web services.

ClientVersion= Root directory for files provided for updating the GUI client by "Version-Check".

BorPict= Storage for borrower pictures.

Upload= Target directory for files uploaded via Web services.

GuiService= Directory containing menus and help texts for GUI services. The respective function originated from Aleph is not in use for Alephino.

Objects= Directory that shall be used for starting point (root) of the Alephino repository for digital objects. To make the objects available as URLs afterwards, this directory has to be mapped to an alias name by means of the respective Web server. With Apache the directive "Alias" is appropriate, customers who are using Microsoft IIS have to define a "virtual directory".

Parameter section (Messages)

This section contains references to files covering language dependent messages of the Alephino server. The parameter names comply with the 3-digit language code.

Example:

```
(Messages)
GER = ../etc/message.ger
ENG = ../etc/message.eng
FRE = ../etc/message.fre
```

Parameter section (Texts)

This section contains references to files covering language dependent texts such as naming of master files, search aspects, link aspects, field names, column headers, labels for GUI elements, fixed text elements for letters and other printouts etc. The parameter names comply with the 3-digit language code.

Example:

```
(Texts)
GER = ../etc/mabtext.ger
ENG = ../etc/mabtext.eng
FRE = ../etc/mabtext.fre
```

Parameter section (Pools) and corresponding

This section contains references to local databases (data pools). The given naming from here will be taken for addressing physical database files as well as internal database control definitions.

Example:

```
(Pools)
Name = Marc

(Marc)
Path = ../pools/DDMARC
SaveFile = ../pools/DDMARC.log
OpenMode = U
DaysLastSave = 3
```

Explanation:

For each database entry labeled **Name=** in section **(Pools)** a corresponding parameter section as follows is expected:

Path= Path name of the physical data pool.

SaveFile= Path name for transaction log belonging to the respective data pool. If applied, all updates to the database since the most recent regular save action are recorded here (redundancy). Based on this a defective database can be recovered without any loss of data.

OpenMode= Operation mode of the database:
U = Open for reading and writing.
O = Open for writing (Write-Only).
I = Open for reading (Read-Only).

DaysLastSave= After that period of time since the most recent data save action a warning will be displayed.

Parameter section (PoolSave)

This section contains declarations for the Alephino save proceeding. The values given here can be overridden by invocation parameters of batch- resp. Web services.

Example:

```
(PoolSave)
Directory = ../backup
Confirm = Y
SaveRelease = N
SaveRecover = N
```

Explanation:

Directory= Directory for data pool backup files.

Confirm=Y/N The save action will be performed after explicit confirmation.

SaveRelease=Y/N After successfully performed save action the belonging transaction log is cleaned automatically.

SaveRecover=Y/N After successfully performed restore of a database from backup missed data will be recovered automatically from belonging transaction log.

Parameter section (Circulation)

This section contains definitions for controlling the circulation.

Example:

```
(Circulation)
Storeroom = 00004
Storeroom = 00005
HoldAvail = N
ForceSelfService = Y
```

Explanation:

Storeroom= This is for defining the code that declares an item collection for closed stack. On hold request of those items a special "closed stack" request is performed. The parameter is repeatable, so multiple codes can be declared.

HoldAvail=Y/N By default a hold request is refused as long as at least one item of the

respective title is freely available. (Exception: periodicals).
This behaviour can be overridden by an **Y**. **Default: N**

ForceSelfService=Y/N On self loaning and self returning via Web OPAC the regular checkings can be suppressed to force priorization of self service actions.

Blockdate= This function creates temporary borrower blockings depending on actual overdues. This can be used as an alternative to the regular claiming proceeding to exclude borrowers temporarily from library patrons list.
There are the following methods available:

SUM	Blocking period is equal to the sum over all pending overdues.
OVERLAP	Blocking period is equal to the longest pending overdue.
FIXED:nn	Fixed blocking period (nn days). This value can be overridden by parameter FIXBLOCK in parameter section BorrPeriod.

Parameter section (Acquisition)

This section contains parameters for acquisitions and serials control

Example:

```
(Acquisition)
Currency = EUR
CheckBudget = Y
CreateItem = Y
VendorArrival = Y
ItemDefault = ../etc/itemdef.cfg
SerItemDefault = ../etc/seritemdef.cfg
PublDescription = Volume $V($Y), Issue $I
```

Explanation:

Currency= The code of the basic currency resp. ratio = 1 for currency calculation

CheckBudget=Y/N The system checks for exceeding of budgets on order- and invoice-transactions.

CreateItem=Y/N Item records will already be created on initial order creation.

VendorArrival=Y/N Force "Group check in" list in serials control GUI to be sorted by vendors.

ItemDefault= Path name of a file that contains patterns for items created out of the acquisitions workflow.

SerItemDefault= Path name of a file that contains patterns for items created out of the serials control workflow.

PublDescription= Pattern for composition of the item field "Description" from schedule information.

Parameter section (TreeView)

The section contains parameters for controlling the navigation tree view in all GUI modules.

Example:

```
(TreeView)
MaxLevel = 2
MaxRecords = 100
InvertRecords = Y
```

Explanation:

Due to compatibility reasons with the Alephino GUI are two different methods resp. protocols used for transmission of navigation tree information. The tree view available with the common GUI search function is unlimited with regard to the hierarchy depth that can be walked through. In contrast the navigation tree function of the GUI catalog editor is limited to 800 entries (nodes plus leaves). Depending on the customer specific use of the Alephino database it might be reasonable to limit the number of hierarchy levels and/or records displayed per level. Otherwise some levels will never appear in that tree view because hierarchy levels located in front do already allocate the 800 entries mentioned above.

MaxLevel= Number of hierarchy levels that will be requested from server on tree view initialization.

MaxRecords= Maximum number of records (leaves) that will be displayed per hierarchy level (node).

InvertRecords=Y/N By default Alephino records are listed reversely to their input order with the oldest record on top. With **Y** this order can be reverted the way that the latest record appears on top.

Parameter section (WebApp)

File **alephino.cfg** serves for server- as well as Web service configuration. The parameters listed below are Web service configuration entries read by program **aliadm(.exe)**.

Example:

```
(Webapp)
Counter = ../data/wadcount.txt
MaxConns = 9
Log = ../temp/aliadm.log
Application = ENG
Dateform = MM.DD.YYYY
Timeout = 300
Translate = ../etc/utf2html.trl
MailFrom = alephino@t-online.de
```

Explanation:

Counter= Path name of the access counter file. The file also serves for synchronization of concurrent access to Web services.

MaxConns= Maximum number of simultaneously managed sessions.

Log= Path name for the temporary file where session information is stored.

Application= Code of the initial communication language. More than one communication languages can be defined; this means the one used on start of the application.

DateForm= Pattern for date format.

Timeout= Session timeout (in seconds). Web service sessions not being triggered during this time are closed automatically.

Translate= Path name of a character translation table for conversion from UTF-8 to HTML code.

MailFrom= Sender address for email messages sent from Web services.

ServRoot= Base URL of the Web service. By default this URL is composed automatically using server name and port passed from HTTP server. Sometimes there is a need to override this base URL.

Whoami= URL of the Web service. By default this URL is composed automatically using server name, port number and script name passed from HTTP server. Sometimes it is necessary to override this URL by a fixed one.

Parameter section (MailAuth)

The server does support authentication in combination with email transmission. The following parameters are necessary to configure SMTP-AUTH protocol support.

Example:

```
(MailAuth)
```

```
Method   = PLAIN
User     = postman
Password = topsecret
```

Explanation:

Method= Authentication method. The following values can be used:
LOGIN, PLAIN, CRAM_MD5. It is necessary to check in advance which method is supported by the respective SMTP server.

User= Mail account user

Password= Mail account password (plain text)

Parameter section (WebMessage)

This section covers references to files containing language dependent texts and patterns used by the Web service module. For parameter name the respective 3-digit communication language code is used.

Example:

```
(WebMessage)
GER = ../etc/admimsg.ger
ENG = ../etc/admimsg.eng
POR = ../etc/admimsg.por
```

Parameter section (PageSet)

The section covers references to directories containing language dependent Web pages and page fragments used by the Web service module. For parameter name the respective 3-digit communication language code is used.

Example:

```
(PageSet)
GER = ../htdoc/aliadm_ger
ENG = ../htdoc/aliadm_eng
POR = ../htdoc/aliadm_por
```

Parameter section (Imex)

Contains a collection of default parameter for controlling data export and import proceedings. These defaults can be overridden by invocation parameters when running standalone batch- and Web services.

Example:

```
(Imex)
Language = ENG
Pool = M
Type = ALEPHINO
```

Explanation

Language= 3-digit language code

Pool= External name of a database (data pool)

Type= Type of the data exchange format. Valid for parameter are:

ALEPHINO	Native Alephino format.
MAB2	MAB2 floppy disk format. (Germany and Austria)
MABI	MAB2 ISO format. (Germany and Austria)
ALEPH	Aleph sequential format.
MARC21	MARC21 format.
XMARC21	MARC21 XML format. (MARC XML formatting is also applicable for German standard compliant MAB2 data.)

Parameter sections for exchange formats

Default parameters for controlling data export and import according to special exchange formats. These values can be overridden by invocation parameters when running standalone batch- and Web services.

Examples:

```
(MAB2)
File = TIT
Type = MAB2
ConvIn    = ../etc/mab_alephino
ConvOut   = ../etc/alephino_mab
TranslIn  = dostoext
TranslOut = exttodos
(MARC21)
File = TIT
Type = MARC21
ConvIn    = ../etc/marc_alephino
ConvOut   = ../etc/alephino_marc
TranslIn  = marctoext
```

```
TranslOut = exttomarc
(MABI)
File = TIT
Type = MABI
ConvIn    = ../etc/mab_alephino
ConvOut   = ../etc/alephino_mab
TranslIn  = mabtoext
TranslOut = exttomab
```

Explanation

File= 3-digit code of an Alephino master file.

Type= Type of data exchange format.

ConvIn= Path name of a control file for table driven data conversion performed while data **import**. The conversion with regard to contents into the internal Alephino format is controlled by a file whose name is build from **ConvIn** followed by a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

ConvOut= Path name of a control file for table driven data conversion performed while data **export**. The conversion with regard to contents from internal Alephino- to the respective exchange format is controlled by a file whose name is build from **ConvOut** followed by a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

TranslIn= Translation table for conversion of external character set to internally used UTF-8 for data import.

TransOut= Translation table for conversion from internal character set UTF-8 into respective external character set for data emport and error reporting while data import.

Parameter section (Convert)

The section covers a collection of (optional) parameters controlling data conversion.

Example:

```
(Convert)
ALEPH = ../etc/aleph_alephino
Z-MAB = ../etc/mab_alephino
Z-MRC = ../etc/marc_alephino
CatConv = ../etc/catconv.tab
Consort = ../etc/alephino_aleph
ConvTest = ../temp/convert.dat
```

Explanation:

ALEPH= Path name of the control file for Aleph- to Alephino conversion.
The complete name is build by appending a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

Z-MAB= Path name of the control file for online conversion of data originated from Z39.50 sources providing MAB2 compliant data into Alephino format. The complete name is build by appending a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

Z-MRC= Path name of the control file for online conversion of data originated from Z39.50 sources providing MARC21 compliant data into Alephino format. The complete name is build by appending a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

CatConv= Path name of the configuration file for controlling interactive data import functionality available with the GUI catalog.

Consort= Path name of the control file for online conversion of Alephino data into Aleph compliant format used for data replication within a consortium. Will be used in combination with the upload of item- and holdings information to the central consortium server. The complete name is build by appending a dot and the 3-digit code of the respective Alephino master file in uppercase letters.

ConvTest= Path name of a temporary file containing the result record of the latest data conversion performed by the server. The file can be used for error analysis.

Parameter section (Spell)

The section covers parameters for the spell checker service available with **"Simple search"** in Web-OPAC.

Example:

```
(Spell)
SuggestionURL = http://www.google.com/tbproxy/spell
Params = <?xml version="1.0" encoding="utf-8" ?><spellrequest textalreadyclipped="0"
         ignoredups="0" ignoredigits="1" ignoreallcaps="1"><text>%s</text></spellrequest>
Wait = 1
```

Explanation:

SuggestionURL= URL of a Web service that receives the original search word and replies with a suggestion. The HTTP method to be used depends on the ending of that URL:
If the URL ends with **?%s**, method GET, otherwise POST is used.

Params= Parameter string with placehoder **%s** for the search word.

NonBlock= This parameter is to define the maximum time (in seconds) the server waits for a connection to the remote spell-check service to be established. If present, it has preference to Parameter section (Communication). In the absence of any **NonBlock** parameter the server might be blocked for undefined time.

Wait= This parameter is to define the maximum wait time (in seconds) for server response. **Default: 10 seconds** If present, it has preference to Parameter section (Communication)

Parameter section (IpFilterAPPLICATION)

Access to the Alephino server can be limited by masking IP address(es). The limitation can be applied to certain services resp. applications APPLICATION by adding the respective application code to the filter section name. In those cases special filter sections do overrule general definitions.

Examples:

```
(IpFilter)
Allowed = 10.1.49.\*
(IpfilterALIX)
Allowed = 10.1.49.153
```

Explanation:

Allowed= IP address (IPv4) with access permission. The parameter is repeatable.

Denied= IP address (IPv4) with no access. The parameter is repeatable.

APPLICATION Valid parameter values are:

ALIX	Alephino X-Service (online retrieval and update service via HTTP interface)
CAT	Cataloguing GUI
CIRC	Circulation GUI
ACQ	Acquisition/Serials control GUI
WAD	Web Services
SIP	Self-Check via SIP2
Z39	Z39.50 Server
WWW	Web OPAC
OBJECTS	Visibility of digital material in Web OPAC

IP addresses can be truncated by asterisk * applicable to any of the four octets. Pls. note that because of the special meaning of the asterisk as leading comment character it has to be prepended by backslash.

Parameter section (Facets)

The extraction of facets from search results in Web OPAC is controlled by parameters covered in this section.

Example:

```
(Facets)
Lookup = 1000
Present = 5
Link = SCO
Link = STT
Link = NCO
Link = THS
```

Explanation:

Lookup= Maximum number of records from current result set that will be scanned while facet extraction.

Present= Number of terms per facet category that will be presented, sorted descending by occurrence.

Link= 3-digit code of the link aspect used for facet extraction. Up to 4 facets can be extracted at the same time, hence the parameter is repeatable 4 times.

Parameter section (Covers)

These parameters are for creation of Book covers as virtual fields.

Example:

```
(Covers)
SourceURL = http://images.amazon.com/images/P/%s.01.LZZZZZZZ.gif
MinSize = 1000
Key = AMACOV
AutoISN10 = Y
Wait = 1
NonBlock = 1
```

Explanation:

SourceURL= URL pattern for cover pictures. The placeholder %s is filled by the program with a key created dynamically from current record data.

MinSize= Minimum size of target image files. (Amazon always replies with a one-pixel-image if the targeted URL is not valid.)

Key= Localization format used for key creation.

AutoISN10=Y With that parameter set to **Y** Alephino looks for a book cover using ISBN10 which is automatically calculated from ISBN13 in case that with ISBN13 no cover picture could be found. **Default: N**

NonBlock= This parameter is to define the maximum time (in seconds) the server waits for a connection to the remote cover-image repository to be established. If present, it has preference to Parameter section (Communication). In the absence of any **NonBlock** parameter the server might be blocked for undefined time.

Wait= This parameter is to define the maximum wait time (in seconds) for server response. **Default: 10 seconds** If present, it has preference to Parameter section (Communication)

Parameter section (BibliothecaRFID)

This section covers parameter for communication with Bibliotheca RFID scanners using BiblioMiddleware SOAP interface.

Example:

```
(BibliothecaRFID)
MiddlewareURL = http://10.1.49.132:7999/Middleware
Credential = topsecret
XDebug = ../data/soap.xml
```

Explanation:

MiddlewareURL= URL of the BiblioMiddleware service.

Credential= If the BiblioMiddleware configuration is configured to ask for a password, this is the place to enter it.

XDebug= For debugging purposes the most recent dialog with the BiblioMiddleware server can be recorded. The respective file name has to be entered here.

Wait= This parameter is to define the maximum wait time (in seconds) for server response. **Default: 10 seconds** If present, it has preference to Parameter section (Communication)

NonBlock= This parameter is to define the maximum time (in seconds) the server waits for a connection to the BiblioMiddleware service to be established. If present, it has preference to Parameter section (Communication). In the absence of any **NonBlock** parameter the server might be blocked for undefined time.

Parameter section (SMSGate)

The below parameters are used for sending "Hold-Request-Filled" messages to end users via SMS.

Example:

```
(SMSGate)
GateURL = http://gate.smsprovider.com/sendsms.asp
Params = receiver=%1&sender=YourLibrary&msg=%2&id=user&pw=pass&msgtype=t
Log = ../data/sms.log
```

Explanation:

GateURL= The URL of an SMS gateway. The HTTP method to be used depends on the ending of that URL:

If the URL ends with **?%s**, method **GET**, otherwise **POST** is used.

Params= CGI Parameter-String with placeholders:

%1 = The receivers mobile phone number
%2 = Message text

Log= Path name of a log file (optional).

Wait= This parameter is to define the maximum wait time (in seconds) for server response. **Default: 10 seconds** If present, it has preference to Parameter section (Communication)

NonBlock= This parameter is to define the maximum time (in seconds) the server waits for a connection to the SMS gateway to be established. If present, it has preference to Parameter section (Communication). In the absence of any **NonBlock** parameter the server might be blocked for undefined time.