SFgate

The WWW Gateway for freeWAIS-sf
Edition 1.108, for SFgate 5.1
January 1997

by Norbert Gövert and Ulrich Pfeifer

1 Overview

SFgate is a gateway between the World Wide Web (WWW) and WAIS (Wide Area Information Servers) written in Perl. It is specially suited for usage with freeWAIS-sf, as it supports all its extensions. But all servers speaking the WAIS protocol can be contacted.

SFgate is designed as a CGI script so that it is usable with any WWW server supporting the CGI standard.

You can set up HTML forms as user interfaces to SFgate. This enables you to use the structured fields features of freeWAIS-sf in a simple manner and makes it easy to the user to specify his queries.

SFgate can interface to any number of databases anywhere on the net at the same time. You just have to specify them within the HTML form. To do this you can use every HTML FORM tag.

SFgate can be used to search databases containing WAIS source descriptions ("databases of databases"). The search result is converted to a query form on the fly.

SFgate is self contained. No call to waisq or waissearch is done. The script itself connects the servers. For local databases, it is not even necessary to connect to the WAIS server; databases residing on the same host as the HTTP server can be searched with the local search feature.

1.1 Architecture

```
   +-----+      +-------------------+     +---------+
   |HTML |      |       HTML        |     | HTML    |
   |query|      |     headlines     |     |requested|
   |form |      |       menu        |     |documents|
   +-----+      +-------------------+     +---------+
      |            ^            |              ^
    query          |      Document request     |
      |        headlines        |          documents
      v            |            v              |
 +--------------------------------------------------+
 |                       SFgate                     |
 +--------------------------------------------------+
   |  ^      |  ^      |  ^              |  ^     |  ^
   |  |      |  |      |  |              |  |     |  |
   v  |      v  |      v  |              v  |     v  |
 +------+  +------+  +------+         +------+ +------+
 | WAIS |  |local |  |local |   ...   | WAIS | | WAIS |
 |Server|  |    DB|  |    DB|         |Server| |Server|
 +------+  +------+  +------+         +------+ +------+
```

The most general case is that SFgate functions as a layer between an SFgate application form (WWW) and the WAIS databases queried. SFgate processes a query transmitted from the HTML form via HTTP and directs it to the WAIS server (respective local WAIS databases) which are specified within the form. The WAIS server process the query on their local WAIS databases and give back a ranking of headlines of query matching documents. The different rankings from the different WAIS databases are merged by SFgate and presented via HTTP to the user. Now the user can select documents from the single ranking. This request is piped through SFgate to the according WAIS server which return the selected documents to SFgate. SFgate now presents the documents to the user via HTTP again.

Alternatively the step to present the headlines can be disabled. Instead of presenting the headline overview you can get the documents directly.

2 Requirements and Related Stuff

To run SFgate you need the following pieces of software:

- a HTTP server which calls SFgate
- a perl interpreter
- the freeWAIS-sf distribution
- the Perl module Wais.pm
- SFgate itself
- you may want to use SFproxy, an indexing HTTP proxy

SFgate, the freeWAIS-sf distribution and the Wais Perl module are available per ftp. Please check archie to find a mirror site near to you.

The load on our server is growing out of bounds. We had to close it for the public — for the big files at least. Since all Packets are routed over the EUnet site, it will be faster to get the stuff from there as from ls6-www (‘`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/`’).

If you want to download freeWAIS-sf, SFgate or the Perl Wais module be sure to get the latest patchlevel (indicated by the last number in the revision number).

2.1 HTTP Server

First of all, you need a HTTP server. SFgate is tested with CERN HTTPD and NCSA HTTPD. If you use another HTTP server you should consider that it implements the CGI specification as SFgate uses the CGI environement variables `PATH_INFO`, `QUERY_STRING` and `REQUEST_METHOD`.

2.2 Perl

SFgate is written in Perl, so the next thing you need is a perl interpreter, version 5.002 or higher. Take a look at CPAN (Comprehensive Perl Archive Network) for the sources:

(‘`http://www.perl.com/CPAN/`’).

2.3 freeWAIS-sf

Although SFgate can connect to every freeWAIS server, it is best suited for use with the freeWAIS-sf (see section “Overview” in *The freeWAIS-sf Manual*) distribution and its extensions. Get freeWAIS-sf 2.0:

‘`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/freeWAIS-sf/`’

Beside for offering own databases the freeWAIS-sf libraries are needed for linking with the Wais.pm module (see Section 2.4 [The Perl Module Wais.pm], page 4). So just compile them and do not remove the freeWAIS-sf source tree until you didn't link Wais.pm.

2.4 The Perl Module Wais.pm

The Perl module Wais provides access to the freeWAIS-sf libraries. It encapsulates the WAIS protocol and for local databases even the WAIS server. SFgate relies on Wais.pm so before installing SFgate you must install Wais.pm.

Having Perl 5.002 or higher and freeWAIS-sf 2.0 installed it is really easy to install Wais. Look at the included documentation.

Obtain Wais from CPAN (Comprehensive Perl Archive Network):

('`http://www.perl.com/CPAN/authors/id/ULPFR/`').

2.5 SFgate

That's this WWW gateway to WAIS! Get SFgate 5.1:

'`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/SFgate/`'

2.6 SFproxy

SFproxy (see section "Overview" in *The SFproxy Manual*) is an indexing HTTP proxy. You can configure it that it watches all your HTTP requests. If a document of content type *text/html* is sent as response, that document is indexed under the URL given in the request using freeWAIS-sf. This way you get a better hotlist because you can search it.

Another mode of operation is that you feed SFproxy with a list of URLs (e.g. the URLs in your history file). SFproxy then fetches the documents and processes them as described above.

SFproxy comes with SFgate (inclusive its own manual).

2.7 Further Documentation

The SFgate Welcome Page
: '`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/`'

This Manual
: is included in the SFgate distribution in texinfo/postscript/dvi/html format. A standalone postscript version is also available ('`ftp://ftp.germany.eu.net/pub/\`

`infosystems/wais/Unido-LS6/SFgate/SFgate.ps.gz`'). A HTML version is available at '`http://ls6-www.informatik.uni-dortmund.de/\ ir/projects/SFgate/SFgate.html`'.

The Manual of freeWAIS-sf (see section "Overview" in *The freeWAIS-sf Manual*)

- '`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/\ freeWAIS-sf/fwsf.texi.gz`'
- '`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/\ freeWAIS-sf/fwsf.dvi.gz`'
- '`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/\ freeWAIS-sf/fwsf.ps.gz`'
- '`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/\ freeWAIS-sf/fwsf.info.gz`'
- '`http://www.wsc.com/freeWAIS-sf/fwsf_toc.html`'
- '`http://ls6-www.informatik.uni-dortmund.de/ir/projects/freeWAIS-sf/`'

The Manual of SFproxy (see section "Overview" in *The SFproxy Manual*)

is included in the SFgate distribution in texinfo/postscript/dvi/html format. A standalone postscript version is also available ('`ftp://ftp.germany.eu.net/pub/\ infosystems/wais/Unido-LS6/SFgate/`'). A HTML version is available at '`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/SFproxy.html`'.

The freeWAIS-sf Frequently Asked Questions

'`ftp://ftp.germany.eu.net/pub/infosystems/wais/Unido-LS6/\ freeWAIS-sf-faq`'

Mailinglist '`wais@wsct.wsc.com`'

There's a mailinglist for freeWAIS-sf and SFgate: Messages to '`wais@wsct.wsc.com`' will be forwarded to the subscribers. To get on the list send a mail to '`majordomo@wsct.wsc.com`':

```
To: majordomo@wsct.wsc.com

subscribe wais
```

To leave the mailing-list send the same mail, but replace *subscribe* with *unsubscribe*. Please note that e-mail should be sent to '`wais@wsct.wsc.com`', not to '`wais@wsc.com`'!

Newsgroup

See the discussion of WAIS related topics in the newsgroup '`comp.infosystems.wais`'. There's an archiv of this newsgroup: '`http://ls6-www.informatik.uni-dortmund.de/ir/search/ciw.html`'.

3rd International WWW Conference '95

Ulrich Pfeifer, Norbert Fuhr and Tung Huynh, *Searching Structured Documents with the Enhanced Retrieval Functionality of freeWAIS-sf and SFgate*: '`http://www.igd.fhg.de/www/www95/proceedings/papers/47/fwsf/fwsf.html`' (Alternativly you can look up this paper in the *Computer Networks and ISDN Systems* journal **27** (6), 1995.)

freeWAIS-sf 2.x & SFgate 5.x User Manual with Figure and Live Examples

from Leopoldo Saggin ('`lsaggin@civ.bio.unipd.it`'): '`http://www-fog.bio.unipd.it/waishelp/waishlp.html`'.

3 Building and Installing SFgate

Having fulfilled the prerequisites (Perl 5.002 or higher and Wais.pm 2.108 or higher) you have to unpack the distribution. Then you can run the `make` script which determines some system properties, and asks you some questions about the installation. After this the executable can be made, tested and installed. Your last step is to configure your HTTP server, so that it can call SFgate.

3.1 Prerequisites

Since SFgate is written in Perl the first thing you need is a perl interpreter, version 5.002 or higher (see Section 2.2 [Perl], page 3). You don't have it already? So what are you waiting for, it's a really nice and usefull piece of software;-)

Next thing is to install the Perl module Wais.pm (see Section 2.4 [The Perl Module Wais.pm], page 4). Installation really is straightforward, but notice that you need the freeWAIS-sf source tree to link this Perl extension. Look at the README of Wais.pm for installation instructions.

3.2 Unpacking the distribution

'`SFgate-5.1.tar.gz`' is a tar archiv compressed with GNU `gzip`. Unpack it with the following command:

```
gunzip -c SFgate-5.1.tar.gz | tar xvf -
```

The distribution will unpack to the directory '`./SFgate-5.1`'.

3.3 Configure SFgate

The first think you might want to do is to set the perl binary to be used from SFgate. If the correct one is on your path you needn't to do anything. Else you should edit the '`Makefile`' and set the `PERL` variable to the correct value.

Then, typing

```
make
```

the first time starts the configuration for SFgate. First some system properties are determined. Following are a few questions concerning installation directories, HTTP virtual directories and so on. Default values are given in braces.

You might want to reuse your '`config.sh`' from older revisions of SFgate to avoid the question section. To do that just copy the old configuration file to the SFgate directory and issue a

```
make show
```

to have a reinitialization of some values. After this you can proceed with

```
make
make test
make install
```

3.3.1 Questions

3.3.1.1 Installation Directory for Perl Libraries

SFgate comes with some libraries which have to be installed in a directory where Perl can find them. Usually this is determined by Perl itself so it might be a good choice to use the default. But if you can't do installation within the Perl installation tree you can specify another directory.

```
Where should the Perl libraries go
 [/usr/local/ls6/perl/lib/site_perl]?
```

3.3.1.2 Installation Directory for Perl Libraries

With SFgate comes SFproxy (see section "Overview" in *The SFproxy Manual*). Perhaps there will be some more utilities in future SFgate releases. Now you're asked to specify a directory where these utilities should be installed. As in the case of the libraries SFgate makes a good proposition. But you can specify another directory.

```
Where should the Perl utilities go
 [/usr/local/ls6/perl/bin]?
```

3.3.1.3 The Default Directory for WAIS Databases

Using the Wais extension, you can make use of the local search facility (see Section 5.4 [Databases], page 14). To search databases locally, SFgate needs to know the name of the database with the complete path. Here you can enter the default directory of your WAIS databases so that you can specify your databases by name only within your SFgate applications.

```
To simplify the specification of (local) databases in your
SFgate application forms enter the default directory where
SFgate can find the local wais databases. If you don't want
to use a default directory answer with 'none'.
What is the default path to the local wais databases
 [/home/robots/wais/wais-sources]?
```

3.3.1.4 HTTP Proxy

SFgate has a build in HTTP client (see Section 5.7.4 [Direct WWW URLs], page 28) which can use a proxy.

```
The next question is about your desired http proxy.
SFgate has a buildin http client, which can use a proxy.
If you don't have a proxy server or you don't want to
use it, answer 'none'.
What is your http proxy
 [none]?
```

If you want to use a HTTP proxy, specify the port:

```
What's the port of your HTTP proxy
 [80]?
```

3.3.1.5 Place of Your HTML pages

Should be set to the directory where your HTML pages reside.

```
Where do your html pages reside
 [/home/robots/www/pages]?
```

3.3.1.6 Document Directory for SFgate

Specify the document directory for SFgate. The real Installation directory for the demos, the documentation and some icons will be the concatenation of your HTML page directory and this document directory.

```
Where  - relative to /home/bib2/goevert/pages/ -
should the documentation go
 [SFgate]?
```

3.3.1.7 Virtual Name of the Document Directory

Specify the virtual name of your document directory for SFgate. The virtual name is used within URLs to specify HTML documents, e.g. '`http://your.http.server/SFgate/demo.html`'.

```
Virtual name means the name you want to use in URLs,
for example in 'http://somehost/SFgate/demo.html'
the virtual name of the document directory is '/SFgate'
What is the virtual name of /home/robots/www/pages/SFgate
 [/SFgate]?
```

3.3.1.8 Directory for CGI Scripts

Set this to the directory where your CGI scripts reside and where the executable SFgate should be installed. Later in the installation section you have to announce this directory to your HTTP server as a directory containing executable scripts (if not done already for other CGIs).

```
Where is your real CGI dir
 [/home/robots/www/pages/cgi-bin]?
```

3.3.1.9 Virtual Name of the CGI Directory

As with the virtual name of your SFgate document directory the virtual name for your CGI directory is used in URLs to call CGI Scripts, e.g. '`http://your.http.server/htbin/SFgate`'.

```
What is the virtual name of /home/robots/www/pages/cgi-bin
 [/htbin]?
```

3.3.1.10 Directory for Application Files

For layout purposes you can specify pieces of html code which are inserted in the header respectivly in the footer of every html page created by SFgate (see Section 5.6.1.1 [Page Headers and Footers], page 20). SFgate searches these pieces of html in files residing in the application files directory.

Configuration for attribute mapping in context with heterogeneous databases (see Chapter 6 [Heterogeneous Databases], page 29) is done in this directory, too.

```
In which directory should the application files go
 [/home/robots/www/etc/SFgate]?
```

3.3.1.11 Directory for SFgate.log

Specify the directory in which the logfile '`SFgate.log`' will go (see Section 5.7.2 [Obtain Logging Information], page 27). Take into consideration that your HTTP server should have write permission in that directory.

```
In which directory should the logfile SFgate.log go
 [/home/robots/www/log]?
```

3.3.1.12 Sending a Registration Mail

After having answered all questions concerning the SFgate configuration you are asked to register yourself as an SFgate user. You are prompted to send us a mail, which includes you SFgate configuration. Registration is done just because of our curiosity. A second reason is that we can better do support if you have problems with SFgate and we've already your configuration.

```
To get an idea of how often and where SFgate is used we would like
you to send us a registration mail. Note that this will *never*
become a licensing scheme or some crazy thing like that. It's just
our curiosity.

Send registration mail to sfgate@ls6.informatik.uni-dortmund.de (y/n)[y]?
```

3.3.1.13 Sending a Registration Mail

If you like SFgate you may want to flood us with gifts. The first thing you could do is to send us a picture postcard of your city. We collect and exhibit them on:

'http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/album/'

Send your card to the following address:

```
Norbert Gövert, Ulrich Pfeifer
University of Dortmund
  Computer Science Department, LS VI
August-Schmidt-Strae 12

 D-44221 Dortmund
```

3.3.2 The Configuration File

After you have answered the questions, a 'config.sh' is created which holds all of the configurations. If you want to change anything of the configuration just edit the file or go through the question section once again:

```
make configure
```

Then all your inputs from the prior configuration are shown as defaults to the questions.

If you just want to see your configuration you should do a

```
make show
```

3.4 Make the Executable and Running the Tests

To make the SFgate executable, just type another

```
make
```

Type

```
make test
```

to do some little tests on your SFgate. They connect to the freeWAIS-sf server on '`ls6-www.informatik.uni-dortmund.de`'. If these tests fail, consider that there can be many reasons, e.g. the connected database ('`wais-docs`') could have changed or your machine has not full internet access. If the test output looks adequat (see the files '`test/is.1`' and '`test/is.2`'), you may proceed and do your own tests with '`demo.html`'.

3.5 Installing

If the tests suceeded, install the software:

```
make install
```

The software and demo files are copied to the directories you specified within the configure run.

At this point you just have to configure your HTTP server to make him execute SFgate. Then you can start using SFgate.

3.5.1 Configuring Your CERN HTTPD

If you are running CERN HTTPD, announce SFgate to your server by editing the config file (called '`httpd.conf`' in most installations). If you had set your real CGI directory to '`/example/real/dir/cgi-bin`' and your virtual CGI directory to '`/example/virtual/dir/htbin`' within `Makefile.PL`, the following lines should be in your configuration file:

```
exec /example/virtual/dir/htbin/* /example/real/dir/cgi-bin/*
```

This annouces the '`/example/real/dir/cgi-bin`' directory to your server.

You can get more info from the WWW Consortium ('`http://www.w3c.org/`').

3.5.2 Configuring Your NCSA HTTPD

If you are running the NCSA Server, announce SFgate to your server by editing the config file (called '`srm.conf`' in most installations). If you had set your real CGI directory to '`/example/real/dir/cgi-bin`' and your virtual CGI directory to '`/example/virtual/dir/htbin`' within `Makefile.PL`, the following line should be in your configuration file:

```
ScriptAlias /example/virtual/dir/htbin /example/real/dir/cgi-bin
```

This annouces the '`/example/real/dir/cgi-bin`' directory to your server.

Get more info from NCSA ('`http://www.ncsa.uiuc.edu/Indices/WebTech/`').

4 SFgate Demos

The first thing you can do is to test the demo (see Appendix A [Sample SFgate Application], page 35). You find it under:

'`http://your.http.server/your-SFgate-doc-dir/demo.html`'

When submitting a query, your SFgate is called, which contacts to the freeWAIS-sf server on '`ls6-www.informatik.uni-dortmund.de`'. You can select from three databases and use some of the search configuration options, e.g. language selection, verbose headlines, multiple choice headline menus.

Then you can try a server search (see Section 5.6.4 [WAIS Source descriptions], page 26):

'`http://your.http.server/your-SFgate-doc-dir/index.html`'.

The next step is to build your own SFgate applications in order to offer your databases.

5 Building SFgate Applications

Searching your databases with SFgate means filling out HTML forms to specify queries. So setting up an SFgate Application means to set up a HTML form which provides the user interface to SFgate.

Most of the examples used in the explanations below are taken from the demo (see Appendix A [Sample SFgate Application], page 35).

5.1 Configuring SFgate Searches

With SFgate you get a lot of configuration options for your searches. Most of them belong to the presentation of search results some of them belong to the generation of the results and some make it possible for you to obtain debugging information.

Principle there are two ways to do the configuration. If you use forms you can do the configuration per FORM tags within your forms. On the other hand almost every configuration option has a default value which is hardwired within '`config.pl`'. Editing '`config.pl`' is the second way to configure SFgate searches. Values in '`config.pl`' act as default settings.

In some cases it does not make sense to have configuration options set per forms. Here the only way to change settings is to edit '`config.pl`'. On the other hand it does not make sense to hardwire some configuration options. Here you have to use configuration per forms.

5.2 SFgate Query Syntax

Querying WAIS databases with SFgate is typically done by filling input fields in forms. Let's have a short look at the query syntax to use within these input fields.

Generally speaking you can use every feature of the freeWAIS-sf query syntax (see section "Query Syntax" in *The freeWAIS-sf Manual*). But according to the different possibilities to build up a SFgate query form the SFgate query syntax has been extended. SFgate parses queries in its own syntax to map them on the freeWAIS-sf query syntax.

A query consists of a set of conditions. A condition consists of a fieldspecifier, a predicate and a comparison value if a special field is to be searched. To search the global part of a WAIS database the fieldspecifier is omitted, the predicate can be one of `soundex` and `phonix`.

Query conditions can be combined using boolean operators and/or parentheses. Query conditions can be nested in arbitrary depth.

See Appendix C [The Detailed Query Syntax], page 45 for a more formal view on the query syntax.

5.3 HTML Forms

Online HTML forms are the appropriate user interface to your databases. Forms make it easy to the user to specify his Information need.

We suppose that you've knowledge of setting up HTML forms. Otherwise "Learning by Doing" is the best way to get familiar with forms contacting SFgate. So if you get to a point where you don't know how to continue take a look at the demo files or at other SFgate applications.

SFgate supports the request methods 'GET' and 'POST'.

5.4 Databases

To specify databases, name the used FORM tags `database`. You have to know the name of the server, the port if different to 210 (the default port number for the WAIS protocol) and the name of the database:

```
<INPUT NAME="database"
       TYPE="checkbox"
      VALUE="ls6-www.informatik.uni-dortmund.de:210/demo">
A demo database with scanned images
```

The following leads to the same database since the port is 210:

```
<INPUT NAME="database"
       TYPE="checkbox"
      VALUE="ls6-www.informatik.uni-dortmund.de:210/demo">
A demo database with scanned images
```

In case you are using the Wais module and you want to search local databases (i.e. databases residing on the same host as your HTTP server) you can use the local search facility. Searches will be much faster than without local search.

Use the `local` keyword to specify local databases and provide the full path to the database:

```
<INPUT NAME="database"
       TYPE="checkbox"
      VALUE="local//home/wais/wais-sources/demo">
A demo database with scanned images
```

Note the double slash.

If you set the default database directory within the `Makefile.PL` run to '`/home/wais/wais-sources`' (see Section 3.3.1.3 [The Default Directory for WAIS Databases], page 7), the following would lead to the same database:

```
<INPUT NAME="database"
       TYPE="checkbox"
```

```
      VALUE="local/demo">
A demo database with scanned images
```

5.5 The Query Part

5.5.1 freeWAIS-sf Fields as FORM Tags

To specify freeWAIS-sf fields (see section "Definition of Field and Index Types" in *The freeWAIS-sf Manual*) in forms for use with SFgate, just name the used FORM tags like the fields existing in your WAIS database.

For example, if you have a field `au` for author in your freeWAIS-sf database you can specify in your form:

```
Author: <INPUT TYPE="text" NAME="au">
```

Everything written in the belonging input field will be searched in the author index.

There's one special field within WAIS databases: the default or *global* field. To search this field you have to provide an input field named `text`:

```
Global Field: <INPUT TYPE="text" NAME="text">
```

5.5.2 More than one FORM Tag per Field

If you want more than one input field for one field in the database you can enumerate the fields. For example if you want two author input fields, specify:

```
Author: <INPUT TYPE="text" NAME="au_1">
Author: <INPUT TYPE="text" NAME="au_2">
```

Later in this section we'll show how to choose individual Boolean connectors for each input field.

5.5.3 Date Fields

Note that freeWAIS-sf gives the possibility to index date fields. A date value in an document is represented in the form `yyyymmdd` in the corresponding index.

To query such an field you can setup three input fields for year, month and day. Suppose we have an entry date field in our database called `ed`:

```
entry date:
year (yyyy, yy):
```

```
    <INPUT TYPE="text"
           NAME="ed_year"
           SIZE=4>
  month (dd, d):
    <INPUT TYPE="text"
           NAME="ed_month"
           SIZE=2>
  day (mm, m):
    <INPUT TYPE="text"
           NAME="ed_day"
           SIZE=2>
```

The second possibility is to set up just the year input field:

```
Entry date: <INPUT TYPE="text" NAME="ed_1_year" SIZE=10>
```

(Note that enumeration is also possible with date fields!) In this case the user has various possibilities to specify dates:

- yyyymmdd, yymmdd
- dd.mm.yyyy, dd.mm.yy, dd.m.yyyy, dd.m.yy, d.mm.yyyy, d.mm.yy, d.m.yyyy, d.m.yy
- dd.mm, dd.m, d.mm, d.m
- yyyy-mm-dd, yyyy-mm-d, yyyy-m-dd, yyyy-m-d, yy-mm-dd, yy-mm-d, yy-m-dd, yy-m-d
- mm/dd/yyyy, mm/dd/yy, mm/d/yyyy, mm/d/yy, m/dd/yyyy, m/dd/yy, m/d/yyyy, m/d/yy

5.5.4 Selecting Fields at Run Time

Another possibility to build up the query form is to provide various input fields for which the user has to decide which field is to be searched.

The name of such an field consists of `fieldsel` and an optional enumeration. For one input field you've to provide various FORM tags:

One tag is for the input field itself, the field name suffix is `_content`:

```
<INPUT TYPE="text" NAME="fieldsel_0_content">
```

The second tag is for field selection, the field name suffix is `_name`:

```
<SELECT NAME="fieldsel_1_name">
<OPTION> au
<OPTION> ti
<OPTION> text
</SELECT>
```

Because field names in WAIS databases often are mnemonic, you could provide a more comprehensive description to the user. The first step is to replace the `_name` in the field selection tag with the suffix `_description`:

```
<SELECT NAME="fieldsel_1_description">
<OPTION> author name
<OPTION> title
<OPTION> text
</SELECT>
```

The second step is to tell SFgate to what field the descriptions do belong. Use _name_<field> as suffix where <field> is replaced with the name of the field in the WAIS database:

```
<INPUT TYPE="hidden" NAME="fieldsel_name_au"   VALUE="author name">
<INPUT TYPE="hidden" NAME="fieldsel_name_ti"   VALUE="title">
<INPUT TYPE="hidden" NAME="fieldsel_name_text" VALUE="text">
```

Note that enumeration does not make sense with the field – description mapping.

By the way, if you enumerate the other fields, be sure that the enumeration comes directly after fieldsel_: Right: 'fieldsel_0_description'. Wrong: 'fieldsel_description_0'.

5.5.5 Predicates for Numeric Fields

Numeric database fields can be searched with the following predicates:

- less than <
- less equal than <=
- equal to ==
- greater equal than >=
- greater than >

You can let the user choose the predicate to be used within searches. Take the name of the FORM tag name of the numeric field (including the enumeration if present) and append the suffix _p:

```
<SELECT NAME="py_p">
<OPTION> &lt;
<OPTION> &lt;=
<OPTION> ==
<OPTION> &gt;=
<OPTION> &gt;
</SELECT>
<INPUT TYPE="text" NAME="py" SIZE=4>
```

The same is valid for date fields. If there are field selection within your form, specify predicates as follows:

```
<SELECT NAME="fieldsel_1_description">
<OPTION> volume number
<OPTION> issue number
<SELECT NAME="fieldsel_1_p">
<OPTION> &lt;
```

```
<OPTION> &lt;=
<OPTION> ==
<OPTION> &gt;=
<OPTION> &gt;
</SELECT>
<INPUT TYPE=TEXT NAME="fieldsel_1_content">

<INPUT TYPE="hidden" NAME="fieldsel_name_vo"   VALUE="volume number">
<INPUT TYPE="hidden" NAME="fieldsel_name_no"   VALUE="issue number">
```

5.5.6 Indextypes for Textual Fields

Analogous to specifying predicates you can specify indextypes for text input fields. Just use suffix `_i` instead of `_p`. Here is a simple example:

```
<INPUT TYPE="text" NAME="au" VALUE="fuhr">
<SELECT NAME="au_i">
<OPTION> soundex and plain
<OPTION> plain
<OPTION> soundex
</SELECT>
```

Valid names for the different indextypes are listed in the table below.

plain text text, plain, strikt, genau, exacto

soundex soundex, ähnlich

phonix phonix

5.5.7 Boolean Connection of Query Conditions

If you set up a form with multiple input fields you may want to choose the boolean operator to connect the contents of the belonging input fields to a query. To do this set up a FORM tag named `tie` and set it to the desired boolean operator.

```
Connect fields with
<INPUT TYPE="radio" NAME="tie" VALUE="and">AND
<INPUT TYPE="radio" NAME="tie" CHECKED VALUE="or">OR.
```

Alternativly set the variable `$tie` in '`config.pl`' to the desired operator. The default value of `tie` is *OR*.

The second way to set boolean connectors applies to connecting single query conditions with the rest of the query.

To specify that a single condition must be met for resulting documents take the FORM tag name of the field and append `_and`:

```
The following condition must be met:
publication year
<INPUT TYPE="text" NAME="py_and">
```

The same result is given with an additional FORM tag which name consists of the input field name plus the extension `_tie`:

```
The following condition must be met:
publication year
<INPUT TYPE="text" NAME="py">
<INPUT TYPE="hidden" NAME="py_tie" VALUE="and">
```

This makes it also possible to let the user of the FORM choose the connection for a single input field:

```
publication year
<INPUT TYPE="text" NAME="py">
connect with:
<SELECT NAME="py_tie">
<OPTION> and
<OPTION> or
</SELECT>
```

This mechanism works also for field selection for input fields.

To set the boolean opeator for in-field query conditions either take an input field tag named `tieinternal` or edit the variable `$tieinternal` in '`config.pl`':

```
Connect in-field query conditions with
<INPUT TYPE="radio" NAME="tieinternal" VALUE="and">AND
<INPUT TYPE="radio" NAME="tieinternal" CHECKED VALUE="or">OR.
```

With `tieinternal` set to *AND* a query condition like `author=(norbert fuhr)` would result in `author=(norbert and fuhr)`.

The language modules ('`$SFgate/lib/SFgate/Languages`') contain translations of the boolean operators for the specific language which you might want to use instead of *AND*, *OR* and *NOT* (see Section 5.6.1.2 [Language of SFgate Output], page 21).

5.5.8 Grouping of Query Conditions

To achieve a two dimensional boolean connection of query conditions you can group several input fields and specify how query conditions specified within this group shall be connected to the rest of the query. Take `group` as name to build up groups:

```
<INPUT TYPE="hidden" NAME="group_1" VALUE="ti_1,ti_2,ti_3">
<INPUT TYPE="hidden" NAME="group_2" VALUE="au_1,au_2,au_3">
```

```
title
<INPUT TYPE="text" NAME="ti_1">
<INPUT TYPE="text" NAME="ti_2">
<INPUT TYPE="text" NAME="ti_3">

Connect this group with:
<SELECT NAME="group_1_tie">
  <OPTION> or
  <OPTION> and
</SELECT>

author name
<INPUT TYPE="text" NAME="au_1">
<INPUT TYPE="text" NAME="au_2">
<INPUT TYPE="text" NAME="au_3">

Connect this group with:
<SELECT NAME="group_2_tie">
  <OPTION> or
  <OPTION> and
</SELECT>
```

Note that grouping of fieldselection input fields is also possible. Take the `fieldsel` identifier with the optional enumeration to name a fieldselection input fields in a grouping element.

Groups can be used to specify an indextype for every member of the group. This can be done for group two in the example above as follows:

```
Select indextypes for the author group:
<SELECT NAME="group_2_i">
  <OPTION> text
  <OPTION> soundex
</SELECT>
```

5.6 SFgate Output

SFgate's output are HTML pages (what else?). You can configure the Layout of these pages.

5.6.1 Layout Considerations

5.6.1.1 Page Headers and Footers

Per default SFgate inserts in header and footer of every page a link to a short description of SFgate.

You can change this behaviour by providing pieces of HTML code to be inserted on SFgate pages instead of the default header and footer.

To do this specify a FORM tag named `application` and set this to the name of the application:

```
<INPUT TYPE="hidden" NAME="application" VALUE="demo">
```

If you set the application files directory (within the `Makefile.PL` run) to '`/example/dir`' (see Section 3.3.1.10 [Directory for Application Files], page 9) create a file '`/example/dir/demo_header`' and insert there the HTML code for the header. If you want to provide language specific headers (see Section 5.6.1.2 [Language of SFgate Output], page 21) you should append the language respective, e.g. for english: '`/example/dir/demo_header_English`'. Note the capitalized first letter!

Define a file for the footer analogous: '`/example/dir/demo_footer`' ('`/example/dir/demo_footer_English`').

5.6.1.2 Language of SFgate Output

SFgate can do its output in several languages:

- english
- dutch
- french
- german
- italian
- portuguese
- spanish
- swedish

Thanks to

Mark Koenen <markko@sci.kun.nl>
for providing dutch translations

Dominique Pinse <pinse@vnet.ibm.com> and Martial Chartoire <chartoir@ipnl.in2p3.fr>
for providing french translations

Leopoldo Saggin <lsaggin@civ.bio.unipd.it>
for providing italian translations

Jose Eduardo M. de Barros <deBarros@bicc.uni-bonn.de>
for providing portuguese translations

Javier Simo <jsimo@iepala.gn.apc.org>
for providing spanish translations

Sven Wijk <svenw@sgu.se>
for providing swedish translations

The default language is english. To select another language either needs editing the variable `$language` within '`config.pl`' or selection via a FORM tag named `language`:

```
<SELECT NAME="language">
<OPTION> english
<OPTION> dutch
<OPTION> french
<OPTION> german
<OPTION> italian
<OPTION> portuguese
<OPTION> spanish
<OPTION> swedish
</SELECT>
```

5.6.2 The Headline Menu

If your search yields to results you are presented a headline menu per default. Every headline points to its corresponding document.

5.6.2.1 Verbose Headlines

To obtain verbose headlines set a FORM tag named `verbose` or the variable `$verbose` in '`config.pl`' to `1`. If you choose verbose headlines you obtain to every headline the following meta information concerning the related document:

- name of the database where the document was found
- size of the document in bytes or kilobytes
- type of the document

The default are non verbose headlines, i.e. `verbose` is set to `0`. You just get the text of the headline.

```
<INPUT TYPE="radio" NAME="verbose" CHECKED VALUE="1">verbose headlines
<INPUT TYPE="radio" NAME="verbose" VALUE="0">short headlines.
```

5.6.2.2 Multiple Choice Headline Menu

Per default you can select one document from the headline menu which is fetched directly after selection. To get further documents you have to return to the headline menu to select the next one and so on.

If you want to select more than one document from the headline menu to get the selected documents concatenated to a list set a FORM tag named `multiple` to `1` or edit the variable `$multiple_choice` within '`config.pl`'.

```
<B>Multiple choice headlines?
<INPUT TYPE="radio" NAME="multiple" CHECKED VALUE="1">YES
<INPUT TYPE="radio" NAME="multiple" VALUE="0">NO.
```

Setting `multiple` to `1` yields to an headline menu where you can select documents by checking checkboxes. Pressing the `fetch document` button yields to fetch the checked documents, pressing the `new choice` button clears the checkboxes.

5.6.2.3 Headline Style

You can choose wether you want your headlines to appear in a description list, in a preformatted style or in a table. Set a FORM tag named `listenv` or the variable `$listenv` in '`config.pl`' to `DL` for a description list (this is the default), to `PRE` for the preformatted style or to `TABLE` for a table.

```
<INPUT TYPE="radio" NAME="listenv" CHECKED VALUE="DL">description list
<INPUT TYPE="radio" NAME="listenv" VALUE="PRE">preformatted
<INPUT TYPE="radio" NAME="listenv" VALUE="TABLE">table.
```

5.6.2.4 Maximum Number of Hits

If you want to customize the number of hits transferred at most take a FORM tag named `maxhits` to specify the desired value or edit the variable `$WAISmaxdoc` in '`config.pl`'. The default is `40`.

```
How many hits do you want at most?
<INPUT NAME="maxhits" TYPE="text" VALUE="40" SIZE=2>
```

5.6.2.5 Omit Headline Menu

To pass over the headline menu presentation of a query result set to get the documents found directly set a FORM tag named `directget` to `1` or edit the variable `$direct_get` within '`config.pl`'. The default is presenting a headline menu (`directget` set to `0`).

If `directget` is set to `1`, SFgate skips the headline menu and concatenates all found documents to the result HTML page.

```
<INPUT NAME="directget" TYPE="checkbox" VALUE="1">
```

5.6.2.6 Pointer to Further Matching Documents

If there are more query matching documents than shown in a search result the user has to reformulate his query by increasing the setting for the maximum number of hits (see Section 5.6.2.4 [Maximum Number of Hits], page 23) to see these documents.

Another way is to provide a pointer to further matching documents. This can be done by setting the value of `range` either in the query form or within '`config.pl`'. If `range` has a value greater than zero SFgate tries to fetch so many documents as specified by `maxhits`, beginning with the `range`'th document. So obviously `range` should be set to `1` in your form if you want to provide pointers to further matching documents:

```
<B>Give pointer to further matching documents?</B>
<INPUT TYPE="radio" NAME="range" CHECKED VALUE="1"><B>YES</B>
<INPUT TYPE="radio" NAME="range" VALUE="0"><B>NO.</B>
```

5.6.3 Presenting Documents

SFgate knows three kinds of document presentation.

The first form is presenting a single document which results from following directly a link from the headline menu. The document is shown without the corresponding headline.

The second form results from bypassing the headline menu: the result is a list of documents with their headlines.

The third form results from a multiple choice headline menu. Here the documents are concatenated without their headlines.

5.6.3.1 Converting Documents

Before showing documents SFgate can convert them. Each converter is encapsulated in its own Perl library. Look at the files in '`SFgate/Converters/`' below the Perl library directory you specified within the `Makefile.PL` run (see Section 3.3.1.1 [Installation Directory for Perl Libraries], page 7).

To make use of a converter set up a FORM tag named `convert`. Use the basename of the library in which the converter resides as the value:

```
Conversion of documents:
<input type="radio" name="convert" value="Bibtex"> BibTeX
<input type="radio" CHECKED name="convert" value="Label"> pretty
```

With multiple choice headline menus there's the possibility to select the conversion from the search result page. Configuration for this feature must be done in the SFgate application form. You have to provide names (and optionally descriptions) of one or more converters which should be selectable from the multiple choice headline menu. This is done via an input field tag named `convertm`. The value is consisting of a semicolon seperated list of converter specification. A converter specification consists of the name of the converter (see the `convert` field above) or the name, a comma and a converter description.

```
<INPUT TYPE="hidden" NAME="convertm"
 VALUE="Label,pretty;bibtex,BibTeX">
```

These are the same converters as used in the example above.

The best thing to do to get to know how to write/install your own converters is to look at the converters coming with the distribution. These are the main keypoints on writing a converter:

- The converter library you want to use has to be named like the converter. The library has to be installed in the 'SFgate/Converters/' below the Perl library directory you specified within the Makefile.PL run. It must be encapsulated in a package named exactly like the library.
- Your library must provide a `convert` subroutine which is called by SFgate. It is called with two arguments, the document text itself and the headline. These two items should be converted and returned. If you don't want the WAIS headline to be diplayed along with the converted document just return an empty string instead.
- HTML special characters (&, <, >) should be encoded in text not being HTML code. To do this you can use the function `&::encode_entities`, e.g. `$ntext = &::encode_entities($ntext)` encodes the special chars in `$ntext`.

Here's an example of a conerter doing exactly nothing:

```
package SFgate::Converter::Non;

## ######################################################################
## convert
## ######################################################################
## interface for SFgate to convert one document
##
## (string) $text:     document to convert
## (string) $headline: headline of document to convert
##
## By default every document is printed with its corresponding wais
## headline. If you want to derive another headline just change
## the $headline variable according to your needs.
##
## HTML special characters (&, <, >) should be encoded in text not
## being HTML code. To do this you can use the function
## &::encode_entities, e.g. $ntext = &::encode_entities($ntext)
## encodes the special chars in $ntext.
##
## returns:
## - string: the converted document ($ntext)
## - string: headline of converted document ($headline)
##
sub convert
{
    my($text, $headline) = @_;
    return($text, $headline);
}
```

5.6.3.2 Converting Document Headlines

Besides the possibility to convert documents you can convert headlines before they are displayed within the headline menu. Therefore you must provide a `convert_headline` subroutine within the converter module to be used. According to the specification of a converting module for documents (see Section 5.6.3.1 [Converting Documents], page 24) set up a FORM tag named `converthl`. Use the basename of the library in which the headline converter resides as the value:

```
<input type="hidden" name="converthl" value="Donothing">
```

And that's the way a headline converter works: Your library holding the headline converter (in the example '`SFgate/Converter/Donothing.pm`') must provide the subroutine `convertheadline` which takes two arguments:

`$headline`
: this variable holds the headline text

`$url`
: this variable holds the url referencing the document the headline belongs to

The return value should be the new headline and the corresponding url. Take a look at the '`Donothing`' example:

```
sub convert_headline
{
    my($headline, $url) = @_;
    # here your headline conversions should go...
    return qq[<A HREF="$url">$headline</A>];
}
```

5.6.4 WAIS Source descriptions

If some or all elements of a query result set consist of WAIS source descriptions (see section "Database Description" in *The freeWAIS-sf Manual*), these are converted to a query form on the fly. Within the result set you can select the corresponding databases by checking checkboxes and sending queries to them through filling a textarea.

If there are WAIS source description within the query result the multiple choice headline menu option is disabled.

5.7 Miscellanous

5.7.1 Obtain Debugging Information

If something does not run as expected perhaps debugging information leads to the solution of the problem.

There are two possibilities to get debugging information. The first is to dump the environement of the process running SFgate the second is to get debugging information concerning the execution of SFgate.

5.7.1.1 Dump the Environment

To obtain the environment settings of your executing SFgate, you can set a FORM tag named `dumpenv` to `1` or `yes`. The environement is dumped to an HTML page and the scripts exits.

```
<SELECT NAME="dmpenv">
<OPTION> no
<OPTION> yes
</SELECT>
```

You can't set this option within '`config.pl`'.

5.7.1.2 Debug Information

To get (a lot of) debugging information you can specify a FORM tag named `debug` and set the value either to `on` or to `1`. A value of `0` or off means not to print debugging messages.

If there are problems with SFgate you should try this option.

```
<SELECT NAME="debug">
<OPTION> off
<OPTION> on
</SELECT>
```

The default value for this option is `off`. If you want to set this option in '`config.pl`' edit the variable `$debug`.

5.7.2 Obtain Logging Information

By default SFgate writes a logfile '`SFgate.log`' in the directory you specified within the `Makefile.PL` (see Section 3.3.1.11 [Directory for SFgate.log], page 9) run. If you want to turn logging off, set the variable `$logging` within '`config.pl`' to 0.

Consider that you HTTP server must have write permissions to '`Sfgate.log`'. Else logging is disabled silently.

You can't set this option within a form.

5.7.3 Detexing for German special Characters

On some terminals it's not very easy to generate the german special characters (Umlaute) like ä, ö, ü, ß. To make this more easy you can specify these characters in LaTeX notation, i.e. `"a` for ä, `"o` for ö, `"u` for ü, `"s` for ß. Just set the detex field or edit the `$detex` variable in '`config.pl`':

```
<INPUT NAME="detex" TYPE="hidden" VALUE="1">
```

5.7.4 Direct WWW URLs

With freeWAIS-sf comes the possibility to index HTML pages and put the URL of the documents in the document id instead of an conventional WAIS document id. Additionaly these documents get the document type `URL`. If SFgate has to show the headline of this document it extracts the URL and lets the headline point directly to the HTML document so that no WAIS request is needed to fetch that document.

Instead there are two possibilities. On the one hand SFgate gives as link just the URL so that the HTTP client has to fetch the document if the user requests it. On the other hand SFgate can code the URL so that a user request of the document leads to a call of SFgate, which then has to fetch the document via its own HTTP client.

The first method is the default one. If you want the second method, you have to edit '`config.pl`' and change the value of `directhttp` from `1` to `0` (or set up an FORM tag named `directhttp`).

Even if the first method is the default way to fetch HTTP documents, there are cases where it makes sense to let SFgate fetch such documents.

The first case if a user does document selection via an multiple choice headline menu. The second case is when the headline menue is to be omitted (`directget` is set).

In both cases the result is a page with possibly more than one document concatenated.

6 Heterogeneous Databases

If there is the need to query more than one WAIS database in parallel the one setting up a query interface may be lucky and the fields available in the different databases coincide exactly in each of the databases. But the more common case is that there are different fields in different databases. *The schemas of the databases differ.*

In most cases this is due to different types of the documents stored in the databases, e.g. the one database holds references to literature while another one holds product descriptions.

SFgate 5.1 is suited to handle the heterogeneity of databases holding documents not differing to much in their types, especially for references to literature, like e.g. articles, books, reports.

The task with respect to heterogeneous databases is to find a mapping of the attributes (or field names) used within a given query and the attributes (or field names) available in the different databases.

6.1 Mapping of Attributes

To understand how to use the attribute mapping facility of SFgate one has to know what happens to a given query within SFgate. As pointed out in the previous section SFgate's task is to take the attributes from a given query and map them on the most suitable attributes being part of the database schema for each database to query.

A simple (but insufficient) solution is to rename the attributes within the databases to those used in queries. But this is sufficient only if all databases include the same attributes (on the semantic level) as those used within the query. So strictly speaking this solution does not deal with heterogeneity.

The solution implemented in SFgate 5.1 is based on a predefined set of attributes for use within queries. To do a mapping of attributes there has to be knowledge of how the attributes are related to each other. So we introduced a lattice (see Section 6.2 [Predefined Attribute Lattice], page 30) on these attributes which reflects the specialization relationships between the attributes.

With means of this lattice the mapping process for attributes can be defined with the four operations *equality*, *specialization*, *generalization* and *ignorance* in the following order:

Equality If a query condition refers to a lattice attribute, which is in the schema of the database to query, no mapping is done. The original query condition is taken to be part of the translated query.

Specialization
If a query condition refers to a lattice attribute which is not in the database to query, but there are more special attributes within the database schema, the mapping is done by generating one new query conditions for each of the more special attributes. This is done by replacing the original attribute from the external level with the more special one of the conceptual level. The new query conditions (if there are more than one) are connected with the Boolean OR to one new query condition as part of the translated query.

Generalization
If a query condition refers to an lattice attribute which is not in the database to query and there aren't any attributes being more special within the schema of the database to query, the attribute is mapped on the nearest more general attribute which is part of the database schema.

Ignorance If neither equality nor specialization nor generalization yielded a translation of the original query condition the whole condition is ignored for the translated query of the actual database.

6.2 Predefined Attribute Lattice

The set of attributes used within the lattice is mainly taken from the Scientific and Technical Attribute Set (STAS) hold by CNIDR. STAS defines standard identifiers for referring to searchable and retrievable fields within scientific and technical databases.

KL-ONE defines a *diffs* operator which allows for inheritance on attributes. Using the *diffs* construct a specialization hierarchy has been introduced on a (small) subset of the STAS attributes:

```
TOP
|
|-keywords
|  |
|  |-content
|  |  |
|  |  +-full-text
|  |  |  |
|  |  |  +-title
|  |  |  |  |
|  |  |  |  |-book-title
|  |  |  |  |-article-title
|  |  |  |  +-series-title
|  |  |  |
|  |  |  |-abstract
|  |  |  +-subject-descriptor
|  |  |
|  |  +-journal-title
|  |
|  |-initiator
|  |  |
|  |  |-author-name
|  |  |-editor-name
|  |  |-corporate
|  |  +-conference
|  |
|  +-publisher
|     |
|     |-publisher-name
|     +-publisher-address
|
```

```
|-date
|  |
|  |-entry-date
|  +-publication-date
|
+-meta
   |
   |-issn
   |-isbn
   |-crc
   |-volume
   |-number
   +-edition
```

TOP The top attribute which subsumes all other attributes.

keywords Subsumes attributes with direct regard to the document.

content Information concerning the document content.

full-text Full document text.

title Information about the title of an document.

book-title Title of the book.

article-title
Title of the article.

series-title Title of the series.

abstract Abstract of a document.

subject-descriptor
Descriptors of an document.

journal-title
Title of the journal.

initiator Information concerning the initiator.

author-name
The name of an author person.

editor-name
The name of an editor person.

corporate The name of a corporation or institution.

conference
The name of a conference.

publisher A publishing organization.

publisher-name
The name of a publishing organization.

publisher-address
The address of a publishing organization.

date Date information.

entry-date The date a record is added to the database.

publication-date
The date when a document has been published.

meta Additional information.

issn International standard serials number.

isbn International standard book number.

crc Classification according to the ACM Computing Reviews Classification System.

volume Volume number of a journal.

number Number of a journal/serials.

edition Edition of a document.

6.3 Attribute Mapping in Practice

Now, what is to do to use SFgate's heterogeneous database handling facility? First of all SFgate must be told to use it. So set up a FORM tag named `attributes`.

```
<INPUT NAME="attributes" TYPE="hidden" VALUE="1">
```

The next thing is to select a suitable lattice. You need not to use the lattice proposed in the previous section. Feel free to build up your own lattice. If not using the proposed one (i.e. file '`$SFgate/lattice`' file, which is installed in the application directory (see Section 3.3.1.10 [Directory for Application Files], page 9)) you have to announce your lattice to SFgate. So set up a FORM tag named `lattice` with the filename of your lattice file as value (including the complete path if you don't want to install it in the application directory):

```
<INPUT NAME="lattice" TYPE="hidden" VALUE="lattice">
```

Now that you have a lattice you can create the query part in your form. Specify the input fields using the attribute names from your lattice only.

The next step is to do some configuration on the databases to query. In general field names in WAIS databases are not taken from your lattice, so you've to tell SFgate a mapping of the database fields onto the attributes within the lattice. Furthermore SFgate needs to know the types of the database fields since fields from different databases mapped on one lattice attribute need not to possess the same type.

Let's start with an enumeration of the different possible types:

- text
- stemming
- numeric
- soundex
- phonix

The types (yes, a database field can have more than one type, e.g. *soundex* and *text* for personal names) of an field can be easily derived from the .fmt-file (see section "Building a Format Description" in *The freeWAIS-sf Manual*) used to create the database.

Now how can the knowledge about a database be told to SFgate? This is done via an external file which should reside in the application file directory (see Section 3.3.1.10 [Directory for Application Files], page 9). Instead of the database specification as described in Section 5.4 [Databases], page 14 the value of a `database` FORM tag must contain the name of that file. If the file doesn't reside in the application file directory it must be specified with the complete path.

Specifying only a file name of a database configuration file within a `database` FORM tag makes it necessary to configure server, port and name of the WAIS database within the configuration file itself.

Another point are converters. Different databases contain documents of different types so there might be a need to use different converters for documents resulting from different databases. This is done within database configuration files, too.

The syntax of database configuration files is taken from Perl. The three examples below form a database configuration file for the WAIS database '`bibdb-html`' on server '`ls6.informatik.uni-dortmund.de`'.

Server, port and name of the database are specified as simple perl variables:

```
$server = 'local';
$port   = '210';
$name   = 'bibdb-html';
```

The database fields, their types and their counterparts within the lattice are given in an anonymous hash reference named `$attributes`. The first part of an entry is the name of a datbase field followed by a colon and the list of types, seperated by commas. The second part is the counterpart within the lattice:

```
$attributes = {
    'py:numeric'      => 'publication-date',
    'au:text,soundex' => 'author-name',
    'ti:stemming'     => 'title',
    'cc:text'         => 'crc',
    'jt:stemming'     => 'journal-title',
    'vo:numeric'      => 'volume',
    'no:numeric'      => 'number',
    'global:text'     => 'keywords'
    };
```

Also the (optional) mapping of converters is given as an anonymous hash reference (named `$converter`). The first part of an entry is the converter name used within the form, the second part is the name of the converter to call:

```
$converter = {
    'BIBTEX'          => 'bibtex',
    'PRETTY'          => 'label:',
    'DEFAULT'         => 'label:'
    };
```

6.4 Further References

If you want to learn more about the bacckgrounds of SFgate's attribute mapping idea take a look at:

Norbert Fuhr (1996)
Object-Oriented and Database Concepts for the Design of Networked Information Retrieval Systems (`http://ls6-www.informatik.uni-dortmund.de/\ ir/reports/96/Fuhr-96.html`)

Norbert Gövert (1996)
SFgate and Heterogeneous Databases (`http://ls6-www.informatik.uni-dortmund\ .de/ir/projects/SFgate/heterogeneous.html`)

SFgate and Heterogeneous Databases - Demo
`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/cs_literature.html`

Appendix A Sample SFgate Application

In the following we present a sample SFgate application, taken from the demo within the SFgate distribution.

```
<HTML>

  <HEAD>
    <TITLE>SFgate 5.1: Demo</TITLE>
    <LINK
      HREF="http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/"
      REL="uses" TITLE="SFgate 5.1">
  </HEAD>

  <BODY>

    <A HREF="http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/">
      <IMG ALIGN=TOP SRC="/~goevert/SFgate/SFgate.gif" ALT="SFgate">
    </A>


    <H1>SFgate 5.1: Demo</H1>

    This is a demo to your revision (5.1) of SFgate. For more
    Information look at the
    <A HREF="http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/SFgate.html">
      Manual</A>. Information on
    <A HREF="http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/">
      the latest revision of this software</A> is available.

    <HR>


    <FORM METHOD="POST"
      ACTION="http://ls6-www.informatik.uni-dortmund.de/ir/cgi-bin/SFgate">

      <H2>Select some of the following databases:</H2>

      <DL>
        <DT> <INPUT NAME="database"
            TYPE="checkbox"
            VALUE="ls6-www.informatik.uni-dortmund.de/demo">
          <B>DEMO</B>
        <DD> Demo database with scanned images <BR>
        <DT> <INPUT NAME="database"
            TYPE="checkbox"
            VALUE="ls6-www.informatik.uni-dortmund.de/bibdb-html"
            CHECKED>
          <B>BIBDB</B>
        <DD> Bibliographic references on information retrieval <BR>
```

```
  <DT> <INPUT NAME="database"
      TYPE="checkbox"
      VALUE="bib.informatik.uni-dortmund.de/journals">
    <B>JTOCS</B>
  <DD>Table of contents of some computer science related journals.
    An <A HREF="http://bib.informatik.uni-dortmund.de/\

         journals/journals-toc.html">
      overview of present journals</A> is available.
</DL>

<HR>


<H2>Enter your query:</H2>

<HR>

The easiest way to build up a SFgate form:

<DL>
  <DT> title
  <DD> <INPUT TYPE="text"
      NAME="ti"
      VALUE="probabilistic">
  <DT> author name
  <DD> <INPUT TYPE="text"
      NAME="au"
      VALUE="fuhr">
    <SELECT NAME="au_i">
      <OPTION> soundex and plain
      <OPTION> plain
      <OPTION> soundex
    </SELECT>
  <DT> text
  <DD> <INPUT TYPE="text"
      NAME="text"
      SIZE=40
      VALUE="information and retrieval">
</DL>

<HR>

Two possibilities to specify a date field within a SFgate form
(within a freeWAIS-sf databases date fields are indexed in
yyyymmdd notation):

<DL>
  <DT> entry date
  <DD> <SELECT NAME="ed_p">
    <OPTION> ==
```

```
    <OPTION> &gt;
    <OPTION> &gt;=
    <OPTION> &lt;
    <OPTION> &lt;=
    </SELECT>
    <INPUT TYPE="text"
      NAME="ed_year"
      SIZE=10> <BR>
    specify in one of the following formats:
    <UL>
      <LI> yyyymmdd, yymmdd
      <LI> dd.mm.yyyy, dd.mm.yy, dd.m.yyyy, dd.m.yy,
        d.mm.yyyy, d.mm.yy, d.m.yyyy, d.m.yy
      <LI> dd.mm, dd.m, d.mm, d.m
      <LI> yyyy-mm-dd, yyyy-mm-d, yyyy-m-dd, yyyy-m-d,
         yy-mm-dd, yy-mm-d, yy-m-dd, yy-m-d
      <LI> mm&#47;dd&#47;yyyy, mm&#47;dd&#47;yy,
        mm&#47;d&#47;yyyy, mm&#47;d&#47;yy,
        m&#47;dd&#47;yyyy, m&#47;dd&#47;yy,
        m&#47;d&#47;yyyy, m&#47;d&#47;yy
    </UL>
  <DT> entry date
  <DD> <SELECT NAME="ed_1_p">
    <OPTION> ==
    <OPTION> &gt;
    <OPTION> &gt;=
    <OPTION> &lt;
    <OPTION> &lt;=
    </SELECT>
    year (yyyy, yy):
    <INPUT TYPE="text"
      NAME="ed_1_year"
      SIZE=4>
    month (dd, d):
    <INPUT TYPE="text"
      NAME="ed_1_month"
      SIZE=2>
    day (mm, m):
    <INPUT TYPE="text"
      NAME="ed_1_day"
      SIZE=2>
</DL>

<HR>

You may want to give the user more than one input field for a
freeWAIS-sf field and let him specify the Boolean connection
between the fields via checkboxes or other input
tags. Furthermore you can group any number of fields to give the
user the possibility to choose the boolean connection of the
groups explicitly:
```

```
<DL>
  <DT> title
  <DD>
    <INPUT TYPE="text"
      NAME="ti_1"
      SIZE=10>
    <SELECT NAME="ti_2_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="ti_2"
      SIZE=10>
    <SELECT NAME="ti_3_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="ti_3"
      SIZE=10>
  <DT>
    <SELECT NAME="group_2_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
  <DD>
  <DT> author name
  <DD>
    <INPUT TYPE="text"
      NAME="au_1"
      SIZE=10>
    <SELECT NAME="au_2_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="au_2"
      SIZE=10>
    <SELECT NAME="au_3_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="au_3"
      SIZE=10>
  <DT>
    <SELECT NAME="group_3_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
```

```
  <DD>
  <DT> text
  <DD>
    <INPUT TYPE="text"
      NAME="text_1"
      SIZE=10>
    <SELECT NAME="text_2_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="text_2"
      SIZE=10>
    <SELECT NAME="text_3_tie">
      <OPTION> or
      <OPTION> and
    </SELECT>
    <INPUT TYPE="text"
      NAME="text_3"
      SIZE=10>
</DL>

<INPUT TYPE="hidden" NAME="group_1" VALUE="ti_1,ti_2,ti_3">
<INPUT TYPE="hidden" NAME="group_2" VALUE="au_1,au_2,au_3">
<INPUT TYPE="hidden" NAME="group_3" VALUE="text_1,text_2,text_3">

<HR>

Let the user decide the field name for an input field. Let him
select the Boolean connection of the input fields content to the
rest of the query:

<DL>
  <DT> <SELECT NAME="fieldsel_0_description">
    <OPTION> title
    <OPTION> author name
    <OPTION> text
    </SELECT>
  <DD> <SELECT NAME="fieldsel_0_i">
    <OPTION> soundex and plain
    <OPTION> plain
    <OPTION> soundex
    </SELECT>
    <INPUT TYPE="text" NAME="fieldsel_0_content">
      connect with:
    <SELECT NAME="fieldsel_0_tie">
      <OPTION> and
      <OPTION> or
    </SELECT>
  <DT> <SELECT NAME="fieldsel_1_description">
    <OPTION> publication year
```

```
    <OPTION> volume number
    <OPTION> issue number
    </SELECT>
  <DD> <SELECT NAME="fieldsel_1_p">
    <OPTION> ==
    <OPTION> &gt;
    <OPTION> &gt;=
    <OPTION> &lt;
    <OPTION> &lt;=
    </SELECT>
    <INPUT TYPE="text" NAME="fieldsel_1_content">
      connect with:
    <SELECT NAME="fieldsel_1_tie">
      <OPTION> and
      <OPTION> or
    </SELECT>
</DL>

<INPUT TYPE="hidden" NAME="fieldsel_name_au"   VALUE="author name">
<INPUT TYPE="hidden" NAME="fieldsel_name_ti"   VALUE="title">
<INPUT TYPE="hidden" NAME="fieldsel_name_text" VALUE="text">
<INPUT TYPE="hidden" NAME="fieldsel_name_py"   VALUE="publication year">
<INPUT TYPE="hidden" NAME="fieldsel_name_vo"   VALUE="volume number">
<INPUT TYPE="hidden" NAME="fieldsel_name_no"   VALUE="issue number">

<HR>

You may want to decide for the user which query conditions must
be met. The contents of the following input fields are and'ed to
the rest of the query:

<DL>
  <DT> publication year
  <DD> <SELECT NAME="py_p">
      <OPTION> &gt;=
      <OPTION> &gt;
      <OPTION> ==
      <OPTION> &lt;=
      <OPTION> &lt;
    </SELECT>
    <INPUT TYPE="text"
      NAME="py_and"
      SIZE=4
      VALUE="1995">
  <DT> volume number
  <DD> <SELECT NAME="vo_p">
      <OPTION> ==
      <OPTION> &gt;=
      <OPTION> &gt;
      <OPTION> &lt;=
      <OPTION> &lt;
```

```
    </SELECT>
    <INPUT TYPE="text"
      NAME="vo_and"
      SIZE=4>
  <DT> issue number
  <DD> <SELECT NAME="no_p">
      <OPTION> ==
      <OPTION> &gt;=
      <OPTION> &gt;
      <OPTION> &lt;=
      <OPTION> &lt;
    </SELECT>
    <INPUT TYPE="text"
      NAME="no_and"
      SIZE=4>
</DL>

<HR>

<INPUT TYPE="submit" VALUE="start search">
<INPUT TYPE="reset"  VALUE="reset query">

<HR>


<H2>Specify options:</H2>

<DL>
  <DT> <B>Dump environment?</B>
    <SELECT NAME="dmpenv">
      <OPTION> no
      <OPTION> yes
    </SELECT>
  <DD> This option will dump the environement to an html pages instead
    of processing the query.
  <DT> <B>Debug:</B>
    <SELECT NAME="debug">
      <OPTION> off
      <OPTION> on
    </SELECT>
  <DD> This option will show you some debug informtion.
  <DT> <B>Language:</B>
    <SELECT NAME="language">
      <OPTION> English
      <OPTION> Dutch
      <OPTION> French
      <OPTION> German
      <OPTION> Italian
      <OPTION> Portuguese
      <OPTION> Spanish
      <OPTION> Swedish
```

```
        </SELECT>
      <DD> Choose the language you want to use.
      <DT> <B>Listenvironment</B>
        <INPUT TYPE="radio" NAME="listenv" CHECKED VALUE="DL">
        <B>description list</B>
        <INPUT TYPE="radio" NAME="listenv" VALUE="PRE">
        <B>preformatted</B>
        <INPUT TYPE="radio" NAME="listenv" VALUE="TABLE">
        <B>table.</B>
      <DD> Choose listenvironment for headlines.
      <DT> <B>Multiple choice headline menu?</B>
        <INPUT TYPE="radio" NAME="multiple" CHECKED VALUE="1"><B>YES</B>
        <INPUT TYPE="radio" NAME="multiple" VALUE="0"><B>NO.</B>
      <DT> <B>Connect fields with</B>
        <INPUT TYPE="radio" NAME="tie" VALUE="and"><B>AND</B>
        <INPUT TYPE="radio" NAME="tie" CHECKED VALUE="or"><B>OR</B>.
      <DT> <B>Connect query parts within fields</B>
        <INPUT TYPE="radio" NAME="tieinternal" VALUE="and"><B>AND</B>
        <INPUT TYPE="radio" NAME="tieinternal" CHECKED VALUE="or"><B>OR</B>.
      <DD> You can tie the fields within the formular either with
        <EM>AND</EM> or with <EM>OR</EM>.
      <DT> <B>Show</B>
        <INPUT TYPE="radio" NAME="verbose" CHECKED VALUE="1">
        <B>verbose headlines</B>
        <INPUT TYPE="radio" NAME="verbose" VALUE="0">
        <B>short headlines.</B>
      <DD>
      <DT> <B>Give pointer to further matching documents?</B>
        <INPUT TYPE="radio" NAME="range" CHECKED VALUE="1"><B>YES</B>
        <INPUT TYPE="radio" NAME="range" VALUE="0"><B>NO.</B>
      <DD>
      <DT> <B>How many hits do you want at most?</B>
        <INPUT NAME="maxhits" TYPE="text" VALUE="5" SIZE=2>
      <DD>
      <DT> <INPUT NAME="directget" TYPE="checkbox" VALUE="1">
        <B>Skip headline menu?</B>
    </DL>

    <INPUT NAME="convert" TYPE="hidden" VALUE="Label">

  </FORM>

  <HR>


  <ADDRESS>
    <A HREF="http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/author.html">
      SFgate 5.1</A>
  </ADDRESS>

</BODY>
```

```
</HTML>
```

Appendix B Some SFgate Applications

Following is a list of SFgate applications. If you've problems using or configuring SFgate, you can use them for demonstration purposes.

'`http://ls6-www.informatik.uni-dortmund.de/ir/search/bibdb-html.html`'
: Bibliographic references on information retrieval, databases, human-computer interaction and related fields.

'`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/demo.html`'
: Demo for SFgate 5.1.

'`http://bib.informatik.uni-dortmund.de/SFgate/UNIDOlit.html`'
: Literature at the University of Dortmund.

'`http://bib.informatik.uni-dortmund.de/journals/journals-index.html`'
: References to articels of some computer science journals.

'`http://ls6-www.informatik.uni-dortmund.de/ir/search/www-pages.html`'
: Pages on the HTTP server '`ls6-www.informatik.uni-dortmund.de`'.

'`http://ls6-www.informatik.uni-dortmund.de/ir/search/ciw.html`'
: Archive of '`comp.infosystems.wais`'.

'`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/cs_literature.html`'
: SFgate and heterogeneous databases demo

You find further links to SFgate applications under

'`http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/`'.

Appendix C The Detailed Query Syntax

A query consists of one or more query conditions. They can be connected with *AND*, *OR* or *NOT*. Leaving out the boolean connector means connecting with *OR*. The *NOT* connector takes the meaning of *AND NOT*.

```
           query -> or_expression or_operator query
                    or_expression query
                    or_expression

   or_expression -> and_expression and_operator or_expression
                    and_expression not_operator or_expression
                    and_expression
```

A query condition can be:

a parenthesed query
: query conditions can be nested in arbitrary depth

a field expression
: consists of a field specifier, a predicate and a comparison value/expression

a literal search expression
: which can be denoted with ingle or double quotes

a proximity expression or an atleast expression

an index expression
: consisting of an index predicate like soundex or phonix

a single term

a wildcard term
: denoted with an asterisk ('*')

```
  and_expression -> '(' query ')'
                    field_expression
                    ''' s_literal_expression '''
                    '"' d_literal_expression '"'
                    proximity_expression
                    atleast_expression
                    index_expression
                    term
                    wildcard_term
```

Boolean operators can be specified in various languages:

```
     or_operator -> 'or'
                    'ou'
                    'oder'
                    'oppure'
                    'o'
                    'of'
```

```
and_operator -> 'and'
                'et'
                'und'
                'e'
                'y'
                'en'

not_operator -> 'not'
                'non'
                'nicht'
                'non'
                'no'
                'niet'
```

The class `s_character` consists of all printable characters without the single quote itself ('''), analogous the class `d_character` consists of all printable characters without the single double quote itself ('"').

```
s_literal_expression -> s_character s_literal_expression
                        s_character

d_literal_expression -> d_character d_literal_expression
                        d_character
```

Proximity expressions consist of two operands (terms) and a proximity operator. The proximity operator consists of '`w/`' or '`pre/`' and a number denoting a word distance.

- '`nuclear w/10 waste`' matches in a document if it contains the terms '`nuclear`' and '`waste`' within 10 words.
- '`nuclear pre/10 waste`' matches in a document if it contains the term '`nuclear`' up to 10 words before the term '`waste`'.

`num1` may be any natural number greater than zero.

```
proximity_expression -> term 'pre/' num1 term
                        term 'w/' num1 term
```

Atleast expressions consist of one operand (term) and the `atleast/` operator with a number denoting the minimum occurance frequency of the term in a matching document.

```
atleast_expression -> 'atleast/' num1 term
```

Various indextypes can be used as similarity predicates. They can be specified for single terms as well as for more complex query parts.

```
index_expression -> indextype term
                    indextype_list { query }

  indextype_list -> indextype, indextype_list
```

```
                              indextype

           indextype -> indextype_text
                        indextype_soundex
                        indextype_phonix

      indextype_text -> 'text'
                        'plain'
                        'strikt'
                        'genau'
                        'exacto'

   indextype_soundex -> 'soundex'
                        'ähnlich'

    indextype_phonix -> 'phonix'
```

A term consists of a list of characters (not including the characters '<', '>', '=', '(', ')', '{', '}', '/', ',' and any whitespace which are interpreted as word delimiters). A `wildcard_term` is a term followed by an asterisk ('*').

```
                term -> char_list

       wildcard_term -> char_list '*'

           char_list -> character char_list
                        character
```

Field expressions are used to search within special fields. A `non_field_expression` is the same as query except that a `field_expression` mustn't occur.

```
    field_expression -> field num_predicate num_term
                        field '=' non_field_expression
```

A `num_term` is an integer or a real value. With numeric fields various comparison predicates can be used.

```
       num_predicate -> '<'
                        '<='
                        '='
                        '=='
                        '>='
                        '>'
```

Index

Table of Contents