



TUTORIAL: NEWS DB SEARCH



IMPORTANT NOTICE

This documentation is provided without a fee. The content of this documentation is provided for informational use only, is subject to change without notice, may contain technical inaccuracies or typographical errors, and should not be construed as a commitment of Lursoft IT. Lursoft IT may make improvements or changes in the product described in this documentation at any time without notice.

COPYRIGHT

2005, Lursoft IT. All rights reserved.

Use of this documentation is subject to the following terms:

The content of this documentation may not be altered or edited in any way. Only conversion to other formats is allowed.

You may create a printed copy for your own personal use.

For all other uses, such as selling printed copies or using (parts of) the documentation in another publication, prior written agreement from Lursoft IT is required.

All brand names and product names used in this document are trade names, service marks, trademarks or registered trademarks of their respective owners.

CONTACT INFORMATION

Email: support@siets.net

Website: www.siets.net

TABLE OF CONTENTS

PREFACE.....	5
Audience	5
Typographic Conventions	5
Abbreviations	5
1. TUTORIAL OVERVIEW.....	6
1.1. About SIETS	6
1.2. Tutorial Objectives	6
1.3. Required Reading	6
1.4. Suggested Reading.....	6
2. DEFINING SEARCH REQUIREMENTS.....	8
2.1. Present Situation.....	8
2.2. Goals	8
2.3. Actions	8
3. NEWS DB SEARCH APPLICATION DESIGN.....	9
4. CHOOSING HARDWARE AND INSTALLING SIETS.....	10
4.1. Choosing Hardware	10
4.2. Installing SIETS.....	10
4.2.1. Installing Linux	10
4.2.2. Installing Web-server	11
4.2.3. Installing SIETS	11
5. ADDING SIETS STORAGE	12
5.1. Prerequisites	12
5.2. Objectives	12
5.3. Tutorial Steps.....	12
6. ADDING AND INDEXING DATA	16
6.1. Prerequisites	16
6.2. Objectives	17
6.3. Tutorial Steps.....	17
6.3.1. Dumping Data from Database	17
6.3.2. Importing Data to SIETS	17
7. DEVELOPING SEARCH FORM	20
7.1. Prerequisites	20
7.2. Objectives	20
7.3. Tutorial Steps.....	20
8. ADDING FEATURES.....	25
8.1. Spelling checker.....	25
8.2. Word stemming.....	26
8.3. Similar document search.....	26
8.4. Fielded search	27
9. APPENDIX A: PHP SCRIPTS USED IN TUTORIAL	31

9.1.	Simple Search Form with Added Features	31
9.2.	Advanced Search Form.....	34
9.3.	Includes	40
9.3.1.	lib_siets.inc.....	40
9.3.2.	lib_http.inc	43
9.3.3.	xml_dom.inc	44
9.3.4.	lib_smart.inc	47
9.3.5.	lib_obcmd.inc	48

PREFACE

This preface is an introduction to the SIETS Tutorial: News DB Search. It defines the audience, and lists typographic conventions and abbreviations used throughout the guide.

This tutorial is compliant with the SIETS server version 3.2 or higher and the SIETS Enterprise Manager version 1.0.

This section contains the following topics:

- [Audience](#)
- [Typographic Conventions](#)
- [Abbreviations](#)

Audience

This tutorial is intended for corporate website designers, project managers, or other interested parties that want to quickly learn how to integrate SIETS search in a corporate website.

Typographic Conventions

The following styles and conventions are used in this guide:

Convention	Description
Verdana	Represents command, function, file and directory names, system messages, and command-line commands.
<u>Hyperlink</u>	Represents a hyperlink. Clicking on this field takes you to the identified place.
<u>Source code</u>	Represents code.

Abbreviations

The following abbreviations are used in this guide.

Abbreviation	Description
XML	Extensible markup language.
XSLT	XML stylesheet transformation.
HTTP	Hypertext transport protocol.
SQL	Structured query language.

1. TUTORIAL OVERVIEW

This tutorial is designed to familiarize a new user with all necessary steps to be performed to add SIETS search functionality to a collection of news articles that are stored in a relational database.

This tutorial is based on imaginary but realistic present situation and goals.

For more information on present situation and goals, see [Defining Search Requirements](#).

This tutorial is not designed to document all SIETS features and functionality.

This section contains the following topics:

- [About SIETS](#)
- [Tutorial Objectives](#)
- [Required Reading](#)
- [Suggested Reading](#)

1.1. About SIETS

SIETS is a system for information storage and retrieval. The SIETS system consists of the SIETS server and application programming interface (API) for building information storage and retrieval applications.

The SIETS server is an operational unit that performs information storing and retrieval tasks by executing a predefined set of commands.

SIETS API is used for building applications that are specific and customized according to your company needs.

1.2. Tutorial Objectives

By the end of this tutorial, you will be able to:

- Choose hardware for the SIETS system according to the size and number of records.
- Install SIETS.
- Add and configure the SIETS storage.
- Add data from your database to the SIETS storage.
- Develop a SIETS search form.

1.3. Required Reading

The following documentation supports the tutorial activities:

Title	Description
<i>SIETS Installation Guide</i>	Describes how to install SIETS.

1.4. Suggested Reading

The following SIETS documentation is available:

Tutorial: News DB Search

Title	Description
<i>SIETS Administration and Configuration Guide</i>	Describes the SIETS administration and configuration concepts and contains step-by-step instructions.
<i>SIETS Developer's Guide</i>	Describes SIETS from an application developer's perspective and provides reference material for building customized applications based on SIETS.

2. DEFINING SEARCH REQUIREMENTS

This section describes present situation, defines goals to be achieved, and presents major actions that must be performed to achieve the defined goals.

2.1. Present Situation

There is a collection of news articles stored in a relational database.

The news article collection has no full-text search functionality, or it is of a quite poor quality and needs a lot of effort to keep it updated.

2.2. Goals

The following goals are set:

- To add full-text search functionality to the news collection.
- To reduce effort for keeping the search functionality updated.
- To introduce as few changes to existing infrastructure as possible.

2.3. Actions

There are the following major actions to be performed to achieve the goals set in the previous section:

- Install the SIETS server.
- Gather and index data from the database.
- Develop a web-based search form.

3. NEWS DB SEARCH APPLICATION DESIGN

The following diagram describes how the SIETS server, news database and users are related.

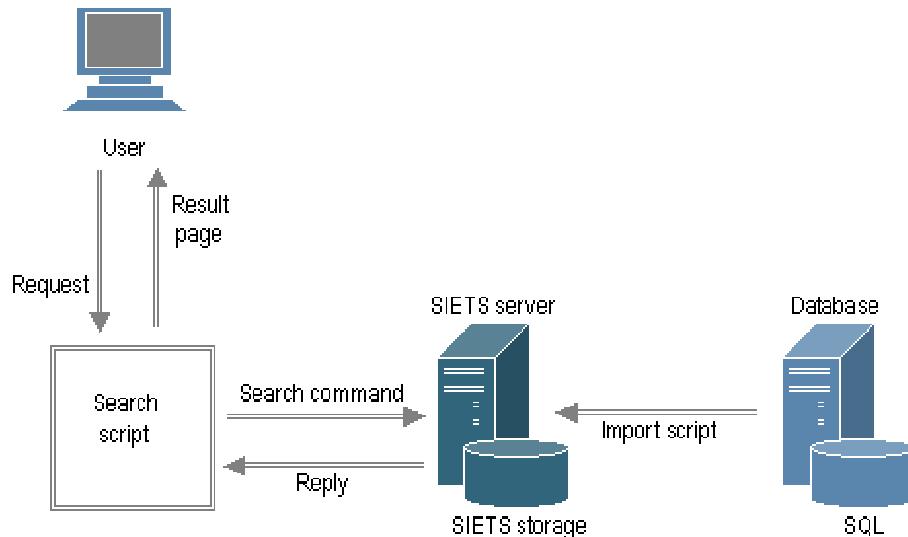


Figure 1: Understanding database search application design

The tasks presented in Figure 1 are explained in the following table:

Task name	Description
Request	A user accesses an information system that contains the search script.
Search command	The search script submits the search command to the SIETS server.
Import script	Data from the database are imported to the SIETS storage using the import script.
Reply	The SIETS server executes the search command and sends reply to the search script.
Result page	The search script displays result page to the user.

4. CHOOSING HARDWARE AND INSTALLING SIETS

This section describes how to choose hardware on which the SIETS system is to be run and how to install SIETS from the SIETS setup that is downloadable from the www.siets.net website and installs the SIETS server and SIETS Enterprise Manager.

In this tutorial, the SIETS server and SIETS Enterprise Manager will be installed on the same computer.

For information on SIETS installation overview, see the *SIETS Installation Guide*, Installation Overview.

4.1. Choosing Hardware

It is recommended to install SIETS server on a separate computer. However, if the size of dataset to be indexed with SIETS is small, the SIETS server can be run together with other applications like web server or database server on the same computer.

The recommended hardware configurations depending on the approximate number of documents are the following:

Number of documents	Total size of documents	Hardware parameters		
		CPU	RAM	Disks
20 000	100 MB	any	512 MB	any
500 000	1 GB	P4	1 GB	any
3 000 000	10 GB	dual Xeon	4 GB	SCSI RAID
> 5 000 000	> 30 GB	The SIETS cluster solution should be considered. Consult SIETS support.		

Note: The parameters provided in the previous table are only for recommendation purposes.

Note: SIETS cluster solutions can be used also for smaller numbers of documents than listed in the previous table. It will provide higher performance on low-cost hardware and provide redundancy or allow handling larger search volumes, > 600 requests per minute.

4.2. Installing SIETS

To install SIETS, there is prerequisite software that needs to be installed before it.

Installing the SIETS server and SIETS Enterprise Manager is the same whether installing SIETS for goals set in this tutorial or for any other scope. Installation is designed as a wizard and the steps are intuitive, also each step is already described in the *SIETS Installation Guide*. Therefore, this section shortly describes each installation part and gives reference to the *SIETS Installation Guide*.

4.2.1. Installing Linux

Currently the SIETS server is available only on Linux operating system.

Prior of installing the SIETS server, Linux must be installed.

As you might know Linux comes with various distributions. SIETS currently has been tested on RedHat, SuSE, Slackware, Mandrake and Debian. However, there should be no problems running SIETS on other distributions.

If you are new to Linux, you can download the ISO image of the SIETS server that is bundled with RedHat Linux 9 from the www.siets.net website. The image installs both: the operating system and SIETS server. The installation from the image is user-friendly, and you will be asked for as little questions as configuring your network parameters.

4.2.2. Installing Web-server

Before installing the SIETS server and SIETS Enterprise Manager, check that web server is installed. A web server is required by SIETS server and SIETS Enterprise Manager to function properly. We recommend using Apache web server, because the SIETS installation detects Apache web-server and integrates within it automatically avoiding additional configuration overhead.

Usually a web server is installed together with an operating system. Check the `httpd` package during Linux installation.

4.2.3. Installing SIETS

You can download the latest SIETS installation version form www.siets.net website. The installation is a shell script that is run from the console. It is interactive and will ask all necessary questions.

After installing SIETS, the web-server must be restarted to apply necessary user rights that are configured by the SIETS installation. To communicate with the SIETS server through UNIX domain sockets those are located in the SIETS storage directory, the user account, which is used to run the web-server, must have an access to the SIETS storage directory.

For detailed information on the installation steps, see the *SIETS Installation Guide*.

5. ADDING SIETS STORAGE

This section describes how to add a new SIETS storage using SIETS Enterprise Manager. You will learn how to add data to the SIETS storage in the next section.

SIETS storage is a data collection for storing SIETS documents in a format that ensures a search is performed very fast.

SIETS Enterprise Manager is an administrative tool, which allows administering and configuring all SIETS system parameters and options.

For more information on SIETS storages and SIETS Enterprise Manager, see the *SIETS Administrator's Guide*, Introduction.

5.1. Prerequisites

To complete steps in this section, the SIETS server must be installed.

5.2. Objectives

In this section you will learn how to add a new SIETS storage and configure it for news database data.

5.3. Tutorial Steps

Perform the following steps:

1. Open the Internet browser.
2. In the **Address** field, enter the following

`http://<server address>/siets/`

where the <server address> is hardware server address on which the SIETS server and SIETS Enterprise Manager is installed.

The SIETS welcome window appears.

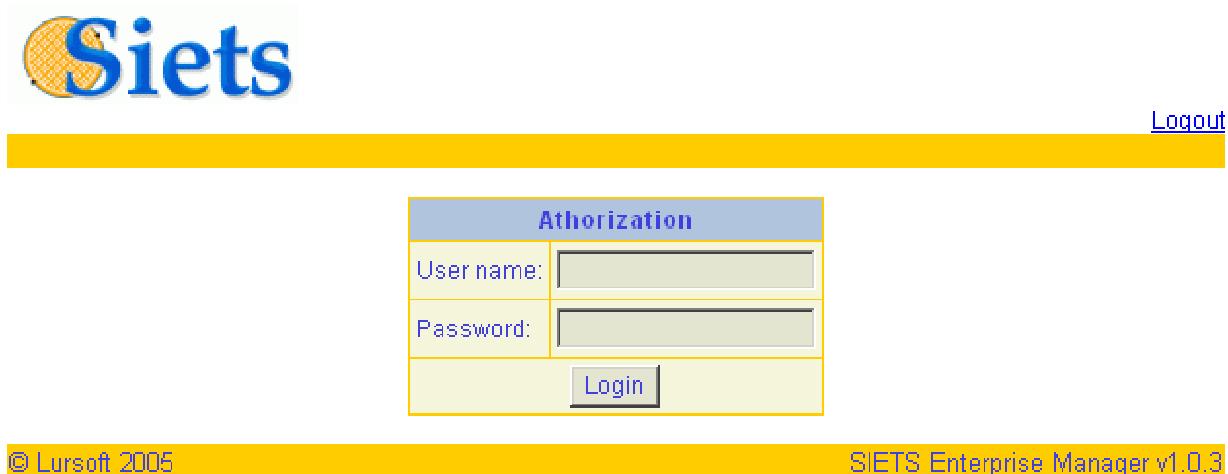


Welcome to SIETS Full Text Search Engine

Figure 2: The SIETS welcome window

3. In the welcome window, click the link.

The SIETS Enterprise Manager authorization window appears.



The screenshot shows the SIETS Enterprise Manager login page. At the top is a yellow header bar with the SIETS logo on the left and a "Logout" link on the right. Below the header is a form titled "Authorization". It contains two input fields: "User name:" and "Password:", both with placeholder text. Below these fields is a blue "Login" button. At the bottom of the page is another yellow footer bar with copyright information on the left and the text "SIETS Enterprise Manager v1.0.3" on the right.

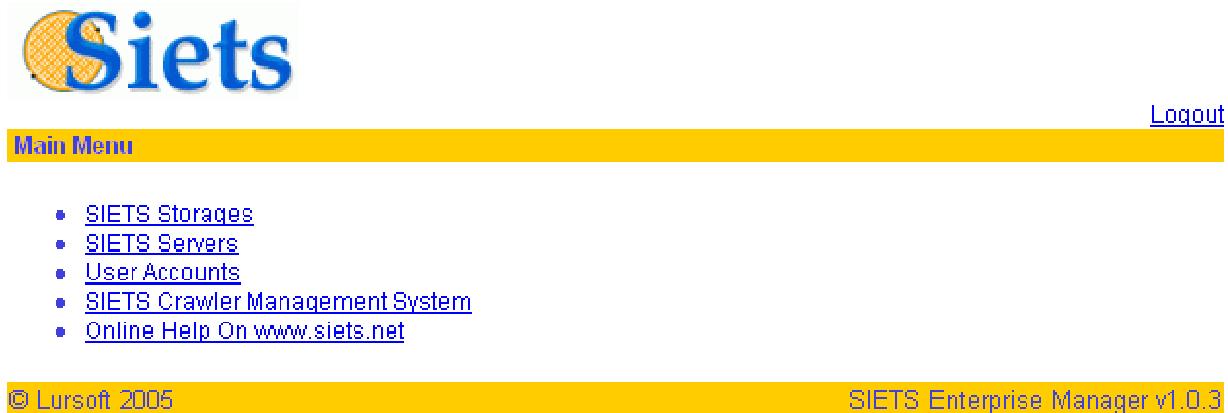
Figure 3: Logging in

4. In the **User name** field, enter 'guest'.
5. In the **Password** field, enter 'guest'.'

For information on administering user accounts, see the *SIETS Administrator's Guide*, Administering SIETS Enterprise Manager User Accounts.

6. Select **Login**.

The **Main Menu** window appears.



The screenshot shows the SIETS Enterprise Manager main menu window. At the top is a yellow header bar with the SIETS logo on the left and a "Logout" link on the right. Below the header is a "Main Menu" button. The main content area contains a bulleted list of links: "SIETS Storages", "SIETS Servers", "User Accounts", "SIETS Crawler Management System", and "Online Help On www.siets.net". At the bottom of the page is another yellow footer bar with copyright information on the left and the text "SIETS Enterprise Manager v1.0.3" on the right.

Figure 4: The Main Menu window

7. Select **SIETS Storages**.

An empty storage list appears.

Name	Status	Number of Servers	Number of Documents	Number of Words: Total/Unique	Action
Refresh	Add Storage				

Figure 5: The SIETS storage list window

8. Select **Add Storage**.

The **Add New Storage** window appears.

Server Host Name	IP	Add to New Storage
localhost	127.0.0.1	<input checked="" type="checkbox"/>
Storage name:	news	
Template:	Default	
Start storage at boot:	<input checked="" type="checkbox"/>	
Storage description:		
<input type="button" value="Create"/> <input type="button" value="Cancel"/>		

Figure 6: Adding SIETS storage

9. To add storage to the SIETS server that has been automatically detected by SIETS Enterprise Manager, select **Add to New Storage** next to the SIETS server IP address.
10. In the **Storage name** field, enter the SIETS storage name, in this case, *news*.
11. In the **Template** drop-down list box, select *Default*.
12. To start the SIETS storage automatically at every boot, select the **Start storage at boot** check box.
13. In the **Storage description** field, enter SIETS storage description of the storage for your own convenience.
14. To finish adding the SIETS storage, click **Create**.

The **SIETS Storage** window appears with the newly added storage in the SIETS storage list with inactive status.

Name	Status	Number of Servers	Number of Documents	Number of Words: Total/Unique	Action
news	Inactive	1		/	<input type="button" value="Start"/>
<input type="button" value="Refresh"/> <input type="button" value="Add Storage"/>					

Figure 7: Viewing newly created SIETS storages list

15. To start the SIETS storage, next to the newly created SIETS storage, select **Start**.

The status of the SIETS storage changes to *Active* and the available action changes to **Stop**.

Tutorial: News DB Search

Name	Status	Number of Servers	Number of Documents	Number of Words: Total/Unique	Action
news	Active	1	0	0 / 0	Stop
Refresh Add Storage					

Figure 8: Starting the SIETS storage

The SIETS storage is up-and-running. No further configuration changes are necessary for news database indexing.

6. ADDING AND INDEXING DATA

This section describes adding data from the news database to the SIETS storage added in the previous section. For this purpose data will be dumped from the database into a comma separated values file and imported to the SIETS storage using a PHP script.

This tutorial assumes imaginary but realistic database structure for news articles.

The MySQL database is used in this tutorial, but SQL statements can be adjusted to other vendors with minor changes.

In this tutorial, it is assumed that data in a database are in the UTF-8 encoding.

The indexing script presented in this section generates a valid XML document from fields of the sample database. The XML document is then imported to the SIETS storage. If you modify the script, for example, in order to add other fields of your database, ensure that a valid XML syntax is preserved, for example, all XML tags are closed.

6.1. Prerequisites

To complete steps in this section, the SIETS storage must be running.

The following database structure that is used as an example in this tutorial:

Table news:

Field	Type	Null	Key	Default	Extra
<code>id</code>	<code>int(11)</code>		PRI	<code>NULL</code>	<code>auto_increment</code>
<code>title</code>	<code>varchar(200)</code>	YES		<code>NULL</code>	
<code>source_id</code>	<code>int(11)</code>	YES		<code>NULL</code>	
<code>description</code>	<code>text</code>	YES		<code>NULL</code>	
<code>published</code>	<code>date</code>	YES		<code>NULL</code>	
<code>lang</code>	<code>char(2)</code>	YES		<code>NULL</code>	

Table source:

Field	Type	Null	Key	Default	Extra
<code>source_id</code>	<code>int(11)</code>		PRI	0	
<code>source_name</code>	<code>varchar(200)</code>	YES		<code>NULL</code>	

The following are SQL statements to create the news and source tables and add some sample records to them:

```
CREATE TABLE source (source_id INT PRIMARY KEY, source_name VARCHAR(200));

INSERT INTO source (source_id, source_name) VALUES
(1, 'Daily Voice'),
(2, 'Morning Issuer');

CREATE TABLE news (id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(200), source_id INT, description TEXT,
published DATE, lang CHAR(2));
```

```
INSERT INTO news (title, source_id, description, published, lang) VALUES
('Hong Kong leader resigns', 1, "Hong Kong's leader Tung Chee-hwa resigned, citing health
reasons for stepping down early after eight turbulent years in office. ", '2005-03-10', 'EN'),
('Boeing interim CEO denies plans to hold to the post', 1, "Boeing Company's President and CEO,
Harry Stonecipher, has stepped down from his positions, after the company asked for his
resignation.", '2005-03-09', 'EN'),
('Scientists issue Malaria warning', 2, "The disease burden is 515 million clinical attacks a year on
the planet. That is quite.", '2005-03-10', 'EN');
```

6.2. Objectives

In this section you will learn how to import data from the database to the SIETS Storage.

6.3. Tutorial Steps

This section contains the following:

- [Dumping Data from Database](#)
- [Importing Data to SIETS](#)

6.3.1. Dumping Data from Database

To dump data from the database, proceed as follows:

1. Use the following SQL statement to retrieve data from the database:

```
SELECT id, title, description, news.source_id, source_name, published, lang FROM news, source
WHERE source.source_id = news.source_id INTO OUTFILE 'news-dump.csv' FIELDS TERMINATED
BY ',' OPTIONALLY ENCLOSED BY "";
```

Syntax of this statement is compatible with MySQL DBMS. If you intend dumping data from other vendor database, refer to it's manual to adjust the SQL statement to produce the same output fields: `id`, `title`, `description`, `source_id`, `source_name`, `published`, and `lang` separated by comma and string fields enclosed in quotes.

Note: The `news-dump.csv` file is located in the databases directory, like `/var/mysql/<db name>`.

2. Copy the dump file to the server where SIETS is installed. You can use `ftp` or `scp` utilities to accomplish that.

6.3.2. Importing Data to SIETS

To import data to the SIETS storage, proceed as follows:

1. Use the following PHP script to import data from the dump file to the SIETS storage.

```
<?php
set_time_limit(0); // no limit to complete
// includes:
require_once("lib_smart.inc");
require_once("lib_siets.inc");
require_once("lib_obcmd.inc");

$FILE_NAME = "news-dump.csv"; // file name of input data
$SIETS_API = "http://127.0.0.1/cgi-bin/siets/api.cgi";
$SIETS_STO = "news"; // storage name to import data
$SIETS_USR = "guest"; // user name for storage
```

Tutorial: News DB Search

```
$SIETS_PAS = "guest"; // password
$DEBUG_DELAY = 0; // delay between inserts
$f = fopen($FILE_NAME,"r");
if (!$f) die("Failed to open $FILE_NAME for reading!\n");

$errors = 0;

while (!feof($f))
{
    $line = fgets($f,102400); //reads at most 102400 bytes from one line
    $line = trim($line);
    while (substr($line,-1)=="\\")
        $line = trim(substr($line,0,-1).fgets($f,102400)); // handle escape sequences
    $valuesx = smart_explode(",",$line,"\"");
    if (count($valuesx)==7) // check correct number of fields
    {
        for ($i=0;$i<count($valuesx);$i++) // handle quote escapes
            $valuesx[$i] = htmlspecialchars(str_replace("\", "", trim($valuesx[$i], "\\")));
        $rep = siets_insert( /* insert to the siets storage */
            $valuesx[0], /* id */
            $valuesx[1], /* title */
            strtotime($valuesx[5]), /* rate - Unix timestamp made from publish date */
            $valuesx[2], /* text */
            "", /* additional info - not neccessary */
            "<source>".$valuesx[3]."</source><publish>".$valuesx[5]."</publish><src_name>".$valuesx[4]."</src_name><title>{$valuesx[1]}</title><lang>{$valuesx[6]}</lang>", /* additional fielded search */
            "", "", "", /* additional not used parameters */
            "UTF-8", /* encoding of the data */
            $SIETS_API, /* API URI */
            $SIETS_STO, /* storage name to index data into */
            $SIETS_USR, /* user name to access storage*/
            $SIETS_PAS /* password to access storage */ );
        if (siets_iserror($rep)) // check for error
        {
            // dump all: requests and replies for first 50 errors
            if ($errors<50)
            {
                $fe = fopen("errors.log.txt","a");
                if ($fe)
                {
                    $qfile = file_get_contents("qfile.xml");
                    $rfile = file_get_contents("rfile.xml");
                    fputs($fe,"==== query ====\n$qfile\n");
                    fputs($fe,"==== reply ====\n$rfile\n==== end ====\n");
                    fclose($fe);
                }
            }
            $errors++;
        }

        obcmd_print($rep."\n");
        sleep($DEBUG_DELAY);
    }
}

fclose($f);
echo "total errors: $errors\n";
?>
```

For all includes and listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

The PHP script presented in this section creates an error file only if there are any errors. The error file contains a dump of requests and replies for transactions that caused errors. For information on the error message structure, see the *SIETS Developer's Guide*, Error Handling.

7. DEVELOPING SEARCH FORM

This section describes developing a search form for the SIETS storage and deploying it in an information system.

7.1. Prerequisites

To complete steps in this section, data must be imported to the *news* storage, and a web-server that is able to execute PHP scripts must be available.

7.2. Objectives

In this section you will learn how to set up the search interface for news articles that are indexed in the SIETS storage.

7.3. Tutorial Steps

To develop a search form, proceed as follows:

1. Log into the web-server where you want to deploy the search form.
2. Find the web root of your web server.

By default, on most distributions, the apache's web root is `/var/www/html`.

3. Change the current directory to the web root.

```
cd /var/www/html
```

4. Make `news-search` directory there.

```
mkdir news-search
```

5. Change the current directory to the `news-search` directory.

```
cd news-search
```

6. Place the `index.php` file with the following content there:

Note: The `index.php` file here is the default file that is read by a Web server when a directory is requested.

```
<?php  
header("Content-type: text/html; charset=UTF-8"); // set charset to UTF-8 using HTTP header<html>  
<head>  
<meta http-equiv="content-type" content="text/html; charset=UTF-8">  
<title>News Search</title>  
</head>  
<body>  
<h2>News Search</h2>  
<form method="get">  
<table border="0">  
<tr>  
<td><b>Query</b></td>  
<td><input type="text" id="query" name="query" size="50" value="<?php echo  
htmlspecialchars(stripslashes($_GET["query"])); ?>" />  
<td><input type="submit" id="search" name="search" value="Search"/></td>
```

Tutorial: News DB Search

```
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="relevance" name="relevance"><?php
if(isset($_GET["relevance"])) echo " checked=\\"checked\\\";?>/>Order results by relevance</td>
<td><br/></td>
</tr>
</table>
<input type="hidden" id="type" name="type" value="search"/>
</form>
<?php
if (!empty($_GET["query"]))
{
    $SIETS_API = "http://127.0.0.1/cgi-bin/siets/api.cgi";
    $SIETS_STO = "news";
    $SIETS_USR = "guest";
    $SIETS_PAS = "guest";
    $PER_PAGE = 10; // results per page

    require_once("lib_siets.inc");
    require_once("xml_dom.inc");

    $page = $_GET["page"];

    $relevance = "";
    $relevance_text = "";
    if (isset($_GET["relevance"]))
    {
        $relevance = "yes";
        $relevance_text = "by relevance ";
    }

    $search_text = "";

    // parse query
    $siets_query = htmlspecialchars(stripslashes($_GET["query"]));

    $res =
siets_search($siets_query,$advanced,$PER_PAGE,$page*$PER_PAGE,$relevance,"","",,$rate_from
,$rate_to,"","","utf-8",$SIETS_API,$SIETS_STO,$SIETS_USR,$SIETS_PAS);
    $real_query = "";
    $realx = array();
    // discover what has been search (after wildcard pattern expansion and stemming if
enabled)
    if (preg_match("/\<real_query\>(.*)\<\/real_query\>/",$res,$realx)>0)
    {
        $real_query = $realx[1];
        $tempx = array();
        if (preg_match("/\{\.(*)\}/",$real_query,$tempx)>0)
        {
            $real_query = $tempx[1];
            $tempx = explode(" ",$real_query);
            $tempx2 = array();
            foreach ($tempx as $word)
            {
                $word = trim($word);
                if (!empty($word))
                    $tempx2[] = $word;
            }
        }
    }
}
```

Tutorial: News DB Search

```
$real_query = implode(" ",$tempx2);
$word_forms = "[in word forms: ".$real_query."]";
}
$search_text = """.htmlspecialchars(stripslashes($_GET["query"]))."";
".$word_forms.$relevance_text;

$xml = @new xml_dom($res);

$from = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'from'}[0]->xml_data[0];
$to = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'to'}[0]->xml_data[0];
$hits = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'hits'}[0]->xml_data[0];
$hitst = $hits;
if ($hitst>1000)
{
    $hitst2 = substr($hitst,0,2);
    while (strlen($hitst2)<strlen($hitst))
        $hitst2 .= "0";
    $hitst = "about ".$hitst2;
}

// display search info
echo "<b>Search for ".$search_text." took ".$xml->{'siets:reply'}[0]->{'siets:seconds'}[0]->xml_data[0]." seconds, found ".$hitst." documents.</b><br/>$alt_text<br/>\n";

// parse result set
if (isset($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}))
{
    foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}[0]->{'document'} as $document)
    {
        $spectags = $document->spectags[0];
        echo "<br/>\n<a href=\"".$spectags->newslink[0]->xml_data[0]."\"
target=_blank"><b>".stripslashes($document->title[0]->xml_data[0])."</b></a><br/>\n";
        if (!empty($document->text[0]->xml_data[0]))
            echo stripslashes($document->text[0]->xml_data[0])."<br/>\n";
        $pubdate = $spectags->adddate[0]->xml_data[0];
        echo "<i><font color=teal>";
        echo date("Y/m/d",$document->rate[0]->xml_data[0]);
        echo " ".htmlspecialchars(urldecode($spectags->src_name[0]->xml_data[0]));
        echo "</font></i>";
        echo "&nbsp;&nbsp;&nbsp;";
        echo "<br/>\n";
    }

    // generate page listing for navigation of Web pages
    $pglist_link = "?";
    foreach ($_GET as $key => $value)
        if ($key!="page")
            $pglist_link .= $key."=".urlencode($value)."&";

    echo "<br/><br/>\n<center>Pages: \n";
    $rpage = (int)floor($from/$PER_PAGE);
    $mpage = (int)floor((($hits-1)/$PER_PAGE));
    if ($rpage>0)
        echo "<a href=\"".$pglist_link."page=".($rpage-1)."\">< &lt; Prev</a> ";
    for ($i=max(0,$rpage-10);$i<=$mpage,$rpage+10);$i++)
    {
        if ($i!=$rpage)
```

```
        echo "<a href=\"".$pglist_link."page=".$i."\">".($i+1)."</a> ";
    else
        echo "<b>".($i+1)."</b> ";
    }
    if ($rpage<$mpage)
        echo "<a href=\"".$pglist_link."page=".($rpage+1).">Next&gt;&gt; </a> ";
    echo "</center>\n";
}
?>
</body>
</html>
```

For all includes and listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

7. Notice the following:

- how the relevance ordering is implemented. HTTP GET parameter is inspected to see if corresponding checkbox is checked then \$relevance variable is set to yes;
 - how the result page is composed. The SIETS server returns results as an XML formatted document, which then is parsed using the XML-DOM parser into tree structure. Then this structure contained in \$xml variable is traveled using foreach statements and for each document tag one entry in list is output;
 - how search info is being displayed. Number of hits is extracted from the SIETS server XML reply as well as time consumed by search operation and the real_query tag value that shows what has been searched;
 - how the page listing is being generated. It uses the returned number of hits and current offset to output page listing with for loop.
8. Access the search form through the Internet browser, URL <http://<server address>/news-search/>

The screenshot shows a web page with a title 'News Search' at the top. Below the title is a search form. The form has a text input field labeled 'Query' containing the placeholder text 'Enter your query here...'. To the right of the input field is a green 'Search' button. Below the input field is a checkbox labeled 'Order results by relevance' which is checked. The entire form is set against a light blue background.

Figure 9: Sample SIETS search form

9. In the sample SIETS search form, enter one or more keywords that are found in the news database, for example, *company*, and select **Search**.

Search results are displayed in the page.

News Search

Query

Order results by relevance

Search for "company" took 0.002 seconds, found 1 documents.

[Boeing interim CEO denies plans to hold to the post](#)
Boeing Company's President and CEO, Harry Stonecipher, has stepped down from his positions, after the company asked for his resignation.
2005/03/09

Pages: 1

Figure 10: Viewing search results

If more results are returned, then results are displayed in several pages and a page listing for navigation through the results are displayed at the bottom of the result page. The number of results per page can be configured using the \$PER_PAGE variable.

If more than 1000 results are returned, then, for performance optimization, an approximate amount of matching documents is estimated, and, in the search results, the amount of matching documents is preceded by the word 'about'.

8. ADDING FEATURES

This section describes adding different SIETS features.

8.1. Spelling checker

To add a spellchecker, insert the following PHP code before the query parsing section in the script listed in [Developing Search Form](#):

```
// spelling check
$alt =
siets_alternatives(htmlspecialchars(htmlspecialchars(stripslashes($_GET["query"]))),","",","",","",",
"utf-8",$SIETS_API,$SIETS_STO,$SIETS_USR,$SIETS_PAS);
$xml = new xml_dom($alt);
$alt_query = "";
$alt_true = false;
foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'alternatives_list'}[0]-
>{'alternatives'} as $alternative)
{
    if (isset($alternative->word))
    {
        $alt_query .= " ".$alternative->word[0]->xml_data[0];
        $alt_true = true;
    }
    else
        $alt_query .= " ".$alternative->to[0]->xml_data[0];
}
if ($alt_true)
{
    $alt_link = "";
    foreach ($_GET as $key => $value)
        if ($key!="query")
            $alt_link .= "&".$key."=".htmlspecialchars($value);
    $alt_text = "Maybe you mean <b><a href=\"?query=".urlencode(trim($alt_query)).$alt_link."\">".htmlspecialchars(trim($alt_query))."</a></b>?<br/>";
}
unset($xml);
```

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

The spellchecking PHP code calls the SIETS alternatives command and presents results in HTML by supplying reasonable alternative word with a similar spelling and higher occurrence rate in data.

Note that the SIETS alternatives command uses statistical analysis of data in the SIETS storage to provide spellchecking. Therefore, it works correctly from the language perspective only if correctly spelled words are imported to the SIETS storage and occurrence of these words is higher than occurrence of those misspelled.

To fine-tune the spelling checker functionality, you can adjust the idif and cr parameters of the alternatives command either through API or change the default values in the SIETS storage configuration.

For more information on the SIETS alternatives command and its parameters, see the *SIETS Developer's Guide*, Alternatives.

8.2. Word stemming

To add the stemming feature, which allows searching different forms of a word, proceed as follows:

1. Insert the following checkbox in the form element in the script listed in [Developing Search Form](#):

```
<tr>
<td><br/></td>
<td><input type="checkbox" id="forms" name="forms"><?php if(isset($_GET["forms"])) echo "checked=\\"checked\\";?>/>Search in word forms</td>
</tr>
```

2. Add the following PHP code after the parse query section in the script listed in [Developing Search Form](#):

```
// word stemming -> enclose query in dollar signs
if (isset($_GET["forms"]))
{
    $siets_query = "$".$siets_query."$";
    $word_forms = "[in word forms] ";
}
```

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

To fine-tune the word stemming functionality, you can adjust the stemming parameters.

For more information on the stemming functionality and its parameters, see the *SIETS Developer's Guide*, Stemming.

8.3. Similar document search

To add the similar search feature, which allows searching similar documents in the SIETS storage to a textual information, which is given directly, or which is contained by a document, proceed as follows:

1. Add the following PHP code to output the [Similar] hyperlink at each document in the result page in the script listed in [Developing Search Form](#):

```
$sim_link = "?";
foreach ($_GET as $key => $value)
    if ($key!="similar" && $key!="page")
        $sim_link .= $key."=".urlencode($value)."&";
echo " <a style=\"color:gray\" href=\"$sim_link相似=$document->id[0]>xml_data[0].\">[Similar]</a>";
```

2. To call the SIETS similar command, add the following if clause in the script listed in [Developing Search Form](#):

```
if (!empty($_GET["similar"]))
{
    // similar document search
    $res =
siets_similar(htmlspecialchars($_GET["similar"]),"",20,5,10,$page*10,"","","utf-
8",$SIETS_API,$SIETS_STO,$SIETS_USR,$SIETS_PAS);
    $search_text = "similar to document #".$_GET["similar"]." ";
}
```

```
...search call...
}
```

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

Note that the similar document search uses artificial intelligence algorithms that are based on statistical analysis of texts. Therefore, the similar search gives good results only for large text collections. You can also try to change the `len` and `quota` parameters (default 20 and 5 in the above sample) to fine-tune similar document search for your dataset. Increasing `len` value gives more diversified results, while increasing `quota` values gives less but more precise results.

For more information on the similar document search functionality and its parameters, see the *SIETS Developer's Guide*, Similar.

8.4. Fielded search

It is possible to add meta-data along plain text to a SIETS document from your database fields. To achieve that, all necessary meta-data must be added to a SIETS document enclosed within XML markup. If documents containing meta-data enclosed within XML markup are indexed to the SIETS storage, then it is possible to build advanced search forms that contain various search fields, each of which searches within a certain XML markup and, thus, specific part of meta-data. In this tutorial, such advanced search form is referred as fielded search.

Content of these XML markup fields are searched as full text, except that search queries for a specific tag must be also enclosed within a specific markup. Thus, a fielded search form reads user input from text fields or drop-down menus and concatenates the input read enclosed with respective markup to the search query.

Searching within multiple meta-data fields can be ensured by combining them in a search query using Boolean operations.

For more information on search within markup and Boolean operations, see the *SIETS Developer's Guide*, Search within Markup and Search Query Syntax, respectively.

In the example presented in this section, the name of an article source and the language of an article are used as meta-data of a SIETS document. The XML tags in a SIETS document are `src_name` and `lang` for the name of an article source and the language, respectively.

To implement fielded search along with the simple search form, proceed as follows:

1. Create the `advanced.php` file that contains script for the advanced search form and supply the following form tag:

```
<form method="get">
<table border="0">
<tr>
<td><b>Query</b></td>
<td><input type="text" id="query" name="query" size="50" value="<?php echo
htmlspecialchars(stripslashes($_GET["query"])); ?>" />
</td>
<tr>
<td><br/></td>
<td><input type="checkbox" id="relevance" name="relevance" <?php
if(isset($_GET["relevance"])) echo " checked=\\"checked\\\";?>/>Order results by relevance</td>
</tr>
<tr>
<td><br/></td>
```

Tutorial: News DB Search

```
<td><input type="checkbox" id="title" name="title"><?php if(isset($_GET["title"])) echo "
checked=\\"checked\\\";?>/>Search in titles only</td>
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="forms" name="forms"><?php if(isset($_GET["forms"])) echo "
checked=\\"checked\\\";?>/>Search in word forms</td>
</tr>
<tr>
<td>Language</td>
<td><select size="1" id="language" name="language">
<?php
$output = "";
$checked = false;
$lang_name = "";
$languagesx = array("en' English", "fr' French", "de' German");
foreach ($languagesx as $language)
{
    $language = trim($language);
    if (!empty($language))
    {
        $code = substr($language,1,2);
        $name = substr($language,5);
        $output .= "<option value=\"$code\"";
        if ($_GET["language"]==$code)
        {
            $checked = true;
            $output .= " selected=\"selected\"";
            if (!empty($name))
                $lang_name = $name;
            else
                $lang_name = $code;
        }
        $output .= ">";
        if (!empty($name))
            $output .= $name;
        else
            $output .= $code;
        $output .= "</option>\n";
    }
}
if ($checked)
    $output = "<option value=\"any\">[Any]</option>\n".$output;
else
    $output = "<option value=\"any\" selected=\"selected\">[Any]</option>\n".$output;
echo $output;
?>
</select></td>
</tr>
<tr>
<td>Source</td>
<td><input type="text" id="source" name="source" size="50" value=<?php echo
htmlspecialchars(stripslashes($_GET["source"])); ?>"></td>
</tr>
<tr>
<td>Date</td>
<td>
&ampnbsp&ampnbsp&ampnbspFrom <input type="text" id="date_from" name="date_from" size="10"
value=<?php echo htmlspecialchars(stripslashes($_GET["date_from"])); ?>">
```

```
&nbsp;&nbsp;&nbsp;To <input type="text" id="date_to" name="date_to" size="10" value="<?php echo htmlspecialchars(stripslashes($_GET["date_to"])); ?>"/>
</td>
</tr>
</table>
<br/>
<input type="hidden" id="type" name="type" value="searchx"/>
<input type="submit" id="searchx" name="searchx" value="Search"/>
</form>
```

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

2. Enter the URL <http://<server address>/news-search/advanced.php> in the Internet browser.

The following form is displayed.

News Search

Query

Order results by relevance

Search in titles only

Search in word forms

Language

Source

Date From To (YYYY/MM/DD)

Figure 11: Viewing fielded search form

3. To parse advanced form input, add the following PHP code and add it to the search query of the SIETS search command in the advanced.php file:

```
$rate_from = "";
$rate_to = "";

$advanced = "";
$advanced_text = "";
if (!empty($_GET["language"])) && $_GET["language"]!="any")
{
    $advanced .= "<lang>".htmlspecialchars($_GET["language"])."</lang>";
    $advanced_text .= "country: ".$lang_name.", ";
}
if (!empty($_GET["source"]))
{
    $advanced .= "
<src_name>".htmlspecialchars(stripslashes($_GET["source"]))."</src_name>";
    $advanced_text .= "source: ".stripslashes($_GET["source"])."";
}
if (!empty($_GET["date_from"]))
{
```

```
$time = strtotime(stripslashes($_GET["date_from"]));
if ($time !== -1)
    $rate_from = $time;
}
if (!empty($_GET["date_to"]))
{
    $time = strtotime(stripslashes($_GET["date_to"]));
    if ($time !== -1)
        $rate_to = $time;
    if ($rate_to == $rate_from && strlen(stripslashes($_GET["date_to"])) <= 10)
        $rate_to += 86399;
}
if (!empty($rate_from) && empty($rate_to))
    $advanced_text .= "published after \"\"".$_GET["date_from"]."\", ";
if (empty($rate_from) && !empty($rate_to))
    $advanced_text .= "published before \"\"".$_GET["date_to"]."\", ";
if (!empty($rate_from) && !empty($rate_to))
    $advanced_text .= "published in
\"".$_GET["date_from"]."\"..\"".$_GET["date_to"]."\"";
```

if (!empty(\$advanced_text))
 \$advanced_text = "(" . substr(\$advanced_text, 0, -2) . ") ";

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

4. Observe that UNIX timestamps from the date range field are calculated. Because the publish date in the UNIX timestamp has been set to the documents rate in the import script, the rate_from and the rate_to parameters of the search command can be used to filter results within a given date interval.
5. To implement the search in a document title only, add the following PHP code that encloses the query in title tags, if the respective checkbox is checked:

```
// search in title only
$tit_only = "";
if (isset($_GET["title"]))
{
    $siets_query = "<title>".$siets_query."</title>";
    $tit_only = "[in titles only] ";
}
```

For complete source listings, see [Appendix A: PHP Scripts Used in Tutorial](#).

Note that this approach works because, in the import script, the title has been once again added to the spectags tag that contains additional meta search info enclosed within the title tag.

9. APPENDIX A: PHP SCRIPTS USED IN TUTORIAL

This appendix presents full listings for sources of PHP scripts used in tutorial.

9.1. Simple Search Form with Added Features

```
<?php
header("Content-type: text/html; charset=UTF-8"); // set charset to UTF-8 using HTTP header
?>
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>News Search</title>
</head>
<body>
<h2>News Search</h2>
<form method="get">
<table border="0">
<tr>
<td><b>Query</b></td>
<td><input type="text" id="query" name="query" size="50" value=<?php echo
htmlspecialchars(stripslashes($_GET["query"])); ?>/>
<td><input type="submit" id="search" name="search" value="Search"/></td>
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="relevance" name="relevance" <?php
if(isset($_GET["relevance"])) echo " checked=\\"checked\\\";?>/>Order results by relevance</td>
<td><br/></td>
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="forms" name="forms" <?php if(isset($_GET["forms"])) echo "
checked=\\"checked\\\";?>/>Search in word forms</td>
</tr>
</table>
<input type="hidden" id="type" name="type" value="search"/>
</form>
<?php
if (!empty($_GET["query"]))
{
    $SIETS_API = "http://195.244.157.207/cgi-bin/siets/api.cgi";
    $SIETS_STO = "news";
    $SIETS_USR = "guest";
    $SIETS_PAS = "guest";
    $PER_PAGE = 10;

    require_once("lib_siets.inc");
    require_once("xml_dom.inc");

    $page = $_GET["page"];

    $relevance = "";
    $relevance_text = "";
    if (isset($_GET["relevance"]))
    {
```

Tutorial: News DB Search

```
$relevance = "yes";
$relevance_text = "by relevance ";
}

$search_text = "";
$alt_text = "";
if (!empty($_GET["similar"]))
{
    // similar document search
    $res =
siets_similar(htmlspecialchars($_GET["similar"]), "", 20, 5, $PER_PAGE, $page * $PER_PAGE, "", "", "utf-8", $SIETS_API, $SIETS_STO, $SIETS_USR, $SIETS_PAS);
    $search_text = "similar to document #".$_GET["similar"]." ";
}
else
{
    // spelling check
    $alt =
siets_alternatives(htmlspecialchars(htmlspecialchars(stripslashes($_GET["query"]))), "", "", "", "", "", "utf-8", $SIETS_API, $SIETS_STO, $SIETS_USR, $SIETS_PAS);
    $xml = new xml_dom($alt);
    $alt_query = "";
    $alt_true = false;
    foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'alternatives_list'}[0]->{'alternatives'} as $alternative)
    {
        if (isset($alternative->word))
        {
            $alt_query .= " ".$alternative->word[0]->xml_data[0];
            $alt_true = true;
        }
        else
            $alt_query .= " ".$alternative->to[0]->xml_data[0];
    }
    if ($alt_true)
    {
        $alt_link = "";
        foreach ($_GET as $key => $value)
            if ($key!="query")
                $alt_link .= "&".$key."=".htmlspecialchars($value);
        $alt_text = "Maybe you mean <b><a href=\"$?query=".urlencode(trim($alt_query)).$alt_link.">".htmlspecialchars(trim($alt_query))."</a></b>?<br/>";
    }
    unset($xml);

    // parse query
    $siets_query = htmlspecialchars(stripslashes($_GET["query"]));
    $word_forms = "";

    // word stemming -> enclose query in dollar signs
    if (isset($_GET["forms"]))
    {
        $siets_query = "$".$siets_query."$";
        $word_forms = "[in word forms] ";
    }

    // search in title only
    $tit_only = "";
}
```

Tutorial: News DB Search

```
if (isset($_GET["title"]))
{
    $siets_query = "<title>".$siets_query."</title>";
    $tit_only = "[in titles only] ";
}

$res =
siets_search($siets_query.$advanced,$PER_PAGE,$page*$PER_PAGE,$relevance,"","",,$rate_from
,$rate_to,"","","utf-8",$SIETS_API,$SIETS_STO,$SIETS_USR,$SIETS_PAS);
$real_query = "";
$realx = array();
// discover what has been search (after wildcard pattern expansion and stemming if
enabled)
if (preg_match("/\<real_query\>(.*)\</real_query\>/",$res,$realx)>0)
{
    $real_query = $realx[1];
    $tempx = array();
    if (preg_match("/\<(.*)\>/",$real_query,$tempx)>0)
    {
        $real_query = $tempx[1];
        $tempx = explode(" ",$real_query);
        $tempx2 = array();
        foreach ($tempx as $word)
        {
            $word = trim($word);
            if (!empty($word))
                $tempx2[] = $word;
        }
        $real_query = implode(" ",$tempx2);
        $word_forms = "[in word forms: ".$real_query."]";
    }
}
$search_text = """.htmlspecialchars(stripslashes($_GET["query"]))."";
".$tit_only.$word_forms.$relevance_text.htmlspecialchars($advanced_text);
}
$xml = @new xml_dom($res);

$from = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'from'}[0]->xml_data[0];
$to = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'to'}[0]->xml_data[0];
$hits = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'hits'}[0]->xml_data[0];
$hitst = $hits;
if ($hitst>1000)
{
    $hitst2 = substr($hitst,0,2);
    while (strlen($hitst2)<strlen($hitst))
        $hitst2 .= "0";
    $hitst = "about ".$hitst2;
}

echo "<b>Search for ".$search_text." took ".$xml->{'siets:reply'}[0]->{'siets:seconds'}[0]->xml_data[0]."
seconds, found ".$hitst." documents.</b><br/>$alt_text<br/>\n";
// parse result set
if (isset($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}))
{
    foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}[0]->{'document'} as
$document)
    {
        $spectags = $document->spectags[0];
```

```

$pop = floatval($spectags->popularity[0]->xml_data[0]);

echo "<br/>\n< a href=\"".$spectags->newslink[0]->xml_data[0]."\"
target=\"_blank\"><b>".stripslashes($document->title[0]->xml_data[0])."</b></a><br/>\n";
if (!empty($document->text[0]->xml_data[0]))
    echo stripslashes($document->text[0]->xml_data[0])."<br/>\n";
$pubdate = $spectags->adddate[0]->xml_data[0];
echo "<i><font color=\"teal\">";
echo date("Y/m/d",$document->rate[0]->xml_data[0]);
echo " ".htmlspecialchars(urldecode($spectags->src_name[0]->xml_data[0]));
echo "</font></i>";
echo "&nbsp;&nbsp;&nbsp;";
$sim_link = "?";
foreach ($_GET as $key => $value)
    if ($key!="similar" && $key!="page")
        $sim_link .= $key."=".urlencode($value)."&";
echo " <a style=\"color:gray\" href=\"".$sim_link."&similar=". $document->id[0]-
>xml_data[0]."\"><Similar></a>";
echo "<br/>\n";
}

// generate page listing
$pglist_link = "?";
foreach ($_GET as $key => $value)
    if ($key!="page")
        $pglist_link .= $key."=".urlencode($value)."&";

echo "<br/><br/>\n<center>Pages: \n";
$rpage = (int)floor($from/$PER_PAGE);
$mpage = (int)floor(($hits-1)/$PER_PAGE);
if ($rpage>0)
    echo "<a href=\"".$pglist_link."page=".($rpage-1)."\">< &lt; Prev</a> ";
for ($i=max(0,$rpage-10);$i<=min($mpage,$rpage+10);$i++)
{
    if ($i!=$rpage)
        echo "<a href=\"".$pglist_link."page=".$i."\"><".($i+1)."</a> ";
    else
        echo "<b>".($i+1)."</b> ";
}
if ($rpage<$mpage)
    echo "<a href=\"".$pglist_link."page=".($rpage+1)."\">> Next&gt;&gt;</a> ";
echo "</center>\n";
}
?>
</body>
</html>

```

9.2. Advanced Search Form

```

<?php
header("Content-type: text/html; charset=UTF-8"); // set charset to UTF-8 using HTTP header
?>
<html>
<head>

```

Tutorial: News DB Search

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8">
<title>News Search</title>
</head>
<body>
<h2>News Search</h2>
<form method="get">
<table border="0">
<tr>
<td><b>Query</b></td>
<td><input type="text" id="query" name="query" size="50" value="<?php echo
htmlspecialchars(stripslashes($_GET["query"])); ?>" />
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="relevance" name="relevance" <?php
if(isset($_GET["relevance"])) echo " checked=\\"checked\\\";?>/>Order results by relevance</td>
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="title" name="title" <?php if(isset($_GET["title"])) echo "
checked=\\"checked\\\";?>/>Search in titles only</td>
</tr>
<tr>
<td><br/></td>
<td><input type="checkbox" id="forms" name="forms" <?php if(isset($_GET["forms"])) echo "
checked=\\"checked\\\";?>/>Search in word forms</td>
</tr>
<tr>
<td>Language</td>
<td><select size="1" id="language" name="language">
<?php
$output = "";
$checked = false;
$lang_name = "";
$languagesx = array("en' English", "fr' French", "de' German");
foreach ($languagesx as $language)
{
    $language = trim($language);
    if (!empty($language))
    {
        $code = substr($language,1,2);
        $name = substr($language,5);
        $output .= "<option value=\"$code\\\"";
        if ($_GET["language"]==$code)
        {
            $checked = true;
            $output .= " selected=\\"selected\\\"";
            if (!empty($name))
                $lang_name = $name;
            else
                $lang_name = $code;
        }
        $output .= ">";
        if (!empty($name))
            $output .= $name;
        else
            $output .= $code;
        $output .= "</option>\n";
    }
}
```

Tutorial: News DB Search

```
}

if ($checked)
    $output = "<option value=\"any\">[Any]</option>\n".$output;
else
    $output = "<option value=\"any\" selected=\"selected\">[Any]</option>\n".$output;
echo $output;
?>
</select></td>
</tr>
<tr>
<td>Source</td>
<td><input type="text" id="source" name="source" size="50" value="<?php echo
htmlspecialchars(stripslashes($_GET["source"])); ?>" /></td>
</tr>
<tr>
<td>Date</td>
<td>
&nbsp;&nbsp;&nbsp;From <input type="text" id="date_from" name="date_from" size="10"
value="<?php echo htmlspecialchars(stripslashes($_GET["date_from"])); ?>" />
&nbsp;&nbsp;&nbsp;To <input type="text" id="date_to" name="date_to" size="10"
value="<?php echo htmlspecialchars(stripslashes($_GET["date_to"])); ?>" />
&nbsp;&nbsp;&nbsp;(YYYY/MM/DD)
</td>
</tr>
</table>
<br/>
<input type="hidden" id="type" name="type" value="searchx"/>
<input type="submit" id="searchx" name="searchx" value="Search"/>
</form>
<?php
{
    $SIETS_API = "http://195.244.157.207/cgi-bin/siets/api.cgi";
    $SIETS_STO = "news";
    $SIETS_USR = "guest";
    $SIETS_PAS = "guest";
    $PER_PAGE = 10;

    require_once("lib_siets.inc");
    require_once("xml_dom.inc");

    $page = $_GET["page"];

    $relevance = "";
    $relevance_text = "";
    if (isset($_GET["relevance"]))
    {
        $relevance = "yes";
        $relevance_text = "by relevance ";
    }

    $rate_from = "";
    $rate_to = "";

    $advanced = "";
    $advanced_text = "";
    if (!empty($_GET["language"])) && $_GET["language"]!="any")
    {
        $advanced .= " <lang>".htmlspecialchars($_GET["language"])."</lang>";
        $advanced_text .= "language: ".$lang_name.", ";
    }
}
```

Tutorial: News DB Search

```
        }
        if (!empty($_GET["source"]))
        {
            $advanced .= "
<src_name>".htmlspecialchars(stripslashes($_GET["source"]))."</src_name>";
            $advanced_text .= "source: ".stripslashes($_GET["source"])." , ";
        }
        if (!empty($_GET["date_from"]))
        {
            $time = strtotime(stripslashes($_GET["date_from"]));
            if ($time !== -1)
                $rate_from = $time;
        }
        if (!empty($_GET["date_to"]))
        {
            $time = strtotime(stripslashes($_GET["date_to"]));
            if ($time !== -1)
                $rate_to = $time;
            if ($rate_to == $rate_from && strlen(stripslashes($_GET["date_to"]))<=10)
                $rate_to += 86399;
        }
        if (!empty($rate_from) && empty($rate_to))
            $advanced_text .= "published after \"\".$_GET["date_from"]."\" , ";
        if (empty($rate_from) && !empty($rate_to))
            $advanced_text .= "published before \"\".$_GET["date_to"]."\" , ";
        if (!empty($rate_from) && !empty($rate_to))
            $advanced_text .= "published in \"\".$_GET["date_from"]."\"..\"\".$_GET["date_to"]."\" , ";
        if (!empty($advanced_text))
            $advanced_text = "(" .substr($advanced_text,0,-2).") ";
    }

    $search_text = "";
    $alt_text = "";
    if (!empty($_GET["similar"]))
    {
        // similar document search
        $res =
siets_similar(htmlspecialchars($_GET["similar"]), "", 20, 5, $PER_PAGE, $page * $PER_PAGE, "", "", "utf-8", $SIETS_API, $SIETS_STO, $SIETS_USR, $SIETS_PAS);
        $search_text = "similar to document #".$_GET["similar"]." ";
    }
    else
    {
        // spelling check
        $alt =
siets_alternatives(htmlspecialchars(htmlspecialchars(stripslashes($_GET["query"]))), "", "", "", "", "", "utf-8", $SIETS_API, $SIETS_STO, $SIETS_USR, $SIETS_PAS);
        $xml = new xml_dom($alt);
        $alt_query = "";
        $alt_true = false;
        if (strlen($query))
            foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'alternatives_list'}[0]->{'alternatives'} as $alternative)
            {
                if (isset($alternative->word))
                {
                    $alt_query .= " ".$alternative->word[0]->xml_data[0];
                    $alt_true = true;
                }
            }
    }
}
```

```

        else
            $alt_query .= ".$alternative->to[0]->xml_data[0];
    }
    if ($alt_true)
    {
        $alt_link = "";
        foreach ($_GET as $key => $value)
            if ($key!="query")
                $alt_link .= "&".$key."=".htmlspecialchars($value);
        $alt_text = "Maybe you mean <b><a href=\"$query=urlencode(trim($alt_query)).$alt_link.\">".htmlspecialchars(trim($alt_query))."</a></b>?<br/>";
    }
    unset($xml);

    // parse query
    $siets_query = htmlspecialchars(stripslashes($_GET["query"]));
    $word_forms = "";

    // word stemming -> enclose query in dollar signs
    if (isset($_GET["forms"]))
    {
        $siets_query = "$".$siets_query.$";
        $word_forms = "[in word forms] ";
    }

    // search in title only
    $tit_only = "";
    if (isset($_GET["title"]))
    {
        $siets_query = "<title>".$siets_query."</title>";
        $tit_only = "[in titles only] ";
    }

    $res =
siets_search($siets_query.$advanced,$PER_PAGE,$page*$PER_PAGE,$relevance,"","",,$rate_from
,$rate_to,"","","utf-8",$SIETS_API,$SIETS_STO,$SIETS_USR,$SIETS_PAS);
    $real_query = "";
    $realx = array();
    // discover what has been search (after wildcard pattern expansion and stemming if
enabled)
    if (preg_match("/\<real_query\>(.*)\</real_query\>/$", $res, $realx) > 0)
    {
        $real_query = $realx[1];
        $tempx = array();
        if (preg_match("/\{\.(*)\}/", $real_query, $tempx) > 0)
        {
            $real_query = $tempx[1];
            $tempx = explode(" ", $real_query);
            $tempx2 = array();
            foreach ($tempx as $word)
            {
                $word = trim($word);
                if (!empty($word))
                    $tempx2[] = $word;
            }
            $real_query = implode(" ", $tempx2);
            $word_forms = "[in word forms: ".$real_query."] ";
        }
    }
}

```

```

        }
    }
    $search_text = """.htmlspecialchars(stripslashes($_GET["query"])).""
".$tit_only.$word_forms.$relevance_text.htmlspecialchars($advanced_text);
}
$xml = @new xml_dom($res);

$from = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'from'}[0]->xml_data[0];
$to = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'to'}[0]->xml_data[0];
$hits = $xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'hits'}[0]->xml_data[0];
$hitst = $hits;
if ($hitst>1000)
{
    $hitst2 = substr($hitst,0,2);
    while (strlen($hitst2)<strlen($hitst))
        $hitst2 .= "0";
    $hitst = "about ".$hitst2;
}

echo "<b>Search for ".$search_text." took ".$xml->{'siets:reply'}[0]->{'siets:seconds'}[0]-
>xml_data[0]." seconds, found ".$hitst." documents.</b><br/>$alt_text<br/>\n";
// parse result set
if (isset($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}))
{
    foreach ($xml->{'siets:reply'}[0]->{'siets:content'}[0]->{'results'}[0]->{'document'} as
$document)
    {
        $spectags = $document->spectags[0];
        $pop = floatval($spectags->popularity[0]->xml_data[0]);

        echo "<br/>\n<a href=\"".$spectags->newslink[0]->xml_data[0]."\"
target=_blank\"><b>".stripslashes($document->title[0]->xml_data[0])."</b></a><br/>\n";
        if (!empty($document->text[0]->xml_data[0]))
            echo stripslashes($document->text[0]->xml_data[0])."<br/>\n";
        $pubdate = $spectags->adddate[0]->xml_data[0];
        echo "<i><font color=\"teal\">";
        echo date("Y/m/d",$document->rate[0]->xml_data[0]);
        echo " ".htmlspecialchars(urldecode($spectags->src_name[0]->xml_data[0]));
        echo "</font></i>";
        echo "&nbsp;&nbsp;&nbsp;";
        $sim_link = "?";
        foreach ($_GET as $key => $value)
            if ($key!="similar" && $key!="page")
                $sim_link .= $key."=".urlencode($value)."&";
        echo " <a style=\"color:gray\" href=\"".$sim_link."&similar=".$document->id[0]-
>xml_data[0]."\"><Similar></a>";
        echo "<br/>\n";
    }

    // generate page listing
    $pglist_link = "?";
    foreach ($_GET as $key => $value)
        if ($key!="page")
            $pglist_link .= $key."=".urlencode($value)."&";

    echo "<br/><br/>\n<center>Pages: \n";
    $rpage = (int)floor($from/$PER_PAGE);
    $mpage = (int)floor(($hits-1)/$PER_PAGE);
    if ($rpage>0)

```

```
echo "<a href=\"".$pglist_link."page=".($rpage-1)."\">&lt;&lt;Prev</a> ";
for ($i=max(0,$rpage-10);$i<=min($mpage,$rpage+10);$i++)
{
    if ($i!=$rpage)
        echo "<a href=\"".$pglist_link."page=".$i."\">".($i+1)."</a> ";
    else
        echo "<b>".($i+1)."</b> ";
}
if ($rpage<$mpage)
    echo "<a href=\"".$pglist_link."page=".($rpage+1)."\">Next&gt;&gt;</a> ";
echo "</center>\n";
}
// -----
function emptynz($text)
{
    return (empty($text) && $text!="0");
}
?>
</body>
</html>
```

9.3. Includes

9.3.1. lib_siets.inc

```
<?php
require_once("lib_http.inc");
define("siets_id_tag","id");

function siets_command($command, $content, $extags, $encoding, $url, $storage, $user, $pass)
{
    $xml = "";
    $xml .= "<?xml version=\"1.0\" encoding=\"$encoding.\"?>\n";
    $xml .= "<siets:request xmlns:siets=\"www.siets.net\">\n";
    $xml .= "<siets:storage>".$storage."</siets:storage>\n";
    $xml .= "<siets:timestamp>".date("Y-m-d H:i:s")."</siets:timestamp>\n";
    $xml .= "<siets:command>".$command."</siets:command>\n";
    $xml .= "<siets:requestid>".date("ydmHis")."</siets:requestid>\n";
    $xml .= "<siets:user>".$user."</siets:user>\n";
    $xml .= "<siets:password>".$pass."</siets:password>\n";
    $xml .= "<siets:reply_charset>".$encoding."</siets:reply_charset>\n";
    if (!empty($extags))
        $xml .= $extags;
    if (!empty($content))
        $xml .="<siets:content>".$content."</siets:content>\n";
    $xml .= "</siets:request>";
    // ----- debug files -----
    $f = @fopen("qfile.xml","w");
    if ($f)
    {
        fputs($f,$xml);
        fclose($f);
    }
    // -----
```

Tutorial: News DB Search

```
$resp = http_data(http_post($url,$xml));
// ----- debug files -----
$f = @fopen("rfile.xml","w");
if ($f)
{
    fputs($f,$resp);
    fclose($f);
}
// -----
return $resp;
}

function siets_insert($id, $title, $rate, $text, $info, $spectags, $exdoc, $excont, $extags,
$encoding, $url, $storage, $user, $pass)
{
    $xml = "";
    $xml .= "<document>\n";
    $xml .= "<".siets_id_tag.">".$id."</".siets_id_tag.">\n";
    $xml .= "<title>".$title."</title>\n";
    $xml .= "<info>".$info."</info>\n";
    $xml .= "<rate>".$rate."</rate>\n";
    $xml .= "<spectags>".$spectags."</spectags>\n";
    if (!empty($exdoc))
        $xml .= $exdoc;
    $xml .= "<text>".$text."</text>\n";
    $xml .= "</document>\n";
    if (!empty($excont))
        $xml .= $excont;
    return siets_command("insert",$xml,$extags,$encoding,$url,$storage,$user,$pass);
}

function siets_search($query, $docs, $offset, $relevance, $case, $from_domain, $rate_from,
$rate_to, $excont, $extags, $encoding, $url, $storage, $user, $pass)
{
    $xml = "";
    $xml .= "<query>$query</query>\n";
    $xml .= "<docs>$docs</docs>\n";
    if(!empty($offset))
        $xml .= "<offset>$offset</offset>\n";
    if (!empty($relevance))
        $xml .= "<relevance>$relevance</relevance>\n";
    if (!empty($case))
        $xml .= "<case_sensitive>$case</case_sensitive>\n";
    if (!empty($from_domain))
        $xml .= "<max_from_domain>$from_domain</max_from_domain>\n";
    if (!empty($rate_from))
        $xml .= "<rate_from>$rate_from</rate_from>\n";
    if (!empty($rate_to))
        $xml .= "<rate_to>$rate_to</rate_to>\n";
    if (!empty($excont))
        $xml .= $excont;
    return siets_command("search",$xml,$extags,$encoding,$url,$storage,$user,$pass);
}

function siets_retrieve($id, $exdoc, $excont, $extags, $encoding, $url, $storage, $user, $pass)
{
    $xml = "";
    $xml .= "<document>\n";
    $xml .= "<".siets_id_tag.">".$id."</".siets_id_tag.">\n";
```

Tutorial: News DB Search

```
if (!empty($exdoc))
    $xml .= $exdoc;
$xml .= "</document>\n";
if (!empty($excont))
    $xml .= $excont;
return siets_command("retrieve",$xml,$extags,$encoding,$url,$storage,$user,$pass);
}

function siets_similar($id, $text, $len, $quota, $docs, $offset, $excont, $extags, $encoding, $url,
$storage, $user, $pass)
{
    $xml = "";
    if (!empty($id))
        $xml .= "<.siets_id_tag.">".$id."</.siets_id_tag.">\n";
    if (!empty($text))
        $xml .= "<text>".$text."</text>\n";
    if (!empty($len))
        $xml .= "<len>".$len."</len>\n";
    if (!empty($quota))
        $xml .= "<quota>".$quota."</quota>\n";
    $xml .= "<docs>$docs</docs>\n";
    if(!empty($offset))
        $xml .= "<offset>$offset</offset>\n";
    if (!empty($excont))
        $xml .= $excont;
    return siets_command("similar",$xml,$extags,$encoding,$url,$storage,$user,$pass);
}

function siets_alternatives($query, $cr, $idif, $h, $excont, $extags, $encoding, $url, $storage,
$user, $pass)
{
    $xml = "";
    $xml .= "<query>".$query."</query>\n";
    if (!empty($cr))
        $xml .= "<cr>".$cr."</cr>\n";
    if (!empty($idif))
        $xml .= "<idif>".$quota."</idif>\n";
    if (!empty($h))
        $xml .= "<h>".$h."</h>\n";
    return siets_command("alternatives",$xml,$extags,$encoding,$url,$storage,$user,$pass);
}

function siets_html($response)
{
    $response = htmlspecialchars($response);
    $response = str_replace("\n","<br/>\n",$response);
    return $response;
}

function siets_iserror($response)
{
    return (strpos($response,"<siets:error>") && strpos($response,"</siets:error>"));
}

function siets_exerror($response, &$code, &$text, &$level, &$source)
{
    $result = siets_iserror($response);
    if ($result)
    {
```

```
$data = array();
preg_match("/<code>(.*)</code>",$response,$data);
$code = $data[1];
preg_match("/<text>(.*)</text>",$response,$data);
$text = $data[1];
preg_match("/<level>(.*)</level>",$response,$data);
$level = $data[1];
preg_match("/<source>(.*)</source>",$response,$data);
$source = $data[1];
}
return $result;
}

?>
```

9.3.2. lib_http.inc

```
$fs = fsockopen($urlx['host'],$urlx['port'],$errno,$error,30);
if ($fs)
{
    fputs($fs,"POST ".$urlx["path"]." HTTP/1.0\r\n");
    fputs($fs,"Host: ".$urlx["host"]."\r\n");
    fputs($fs,"Content-Length: ".strlen($data)."\r\n");
    if (!empty($headers))
        fputs($fs,$headers);
    fputs($fs,"\r\n");
    fputs($fs,$data);
    $reply = "";
    while (!feof($fs))
    {
        $buf = fgets($fs,128);
        $reply .= $buf;
    }
    fclose($fs);
    return $reply;
}
else
    return "[http_post] Error $errno: $error";
}

function http_post_proxy($proxy, $url, $data = "", $headers = "")
{
    $errno = 0; $error = "";
    $urlx = parse_url($url);
    $proxyx = parse_url($proxy);
    if (empty($proxyx['port'])) $proxyx['port'] = 8080;
    $fs = fsockopen($proxyx['host'],$proxyx['port'],$errno,$error,30);
    if ($fs)
    {
        fputs($fs,"POST ".$url." HTTP/1.0\r\n");
        fputs($fs,"Host: ".$urlx["host"]."\r\n");
        fputs($fs,"Content-Length: ".strlen($data)."\r\n");
        if (!empty($headers))
            fputs($fs,$headers);
        fputs($fs,"\r\n");
        fputs($fs,$data);
        $reply = "";
    }
}
```

```
while (!feof($fs))
{
    $buf = fgets($fs,128);
    $reply .= $buf;
}
fclose($fs);
return $reply;
}
else
    return "[http_post_proxy] Error $errno: $error";
}

function http_headers($data)
{
    if (($pos = strpos($data,"r\nr\n"))!==false)
        return substr($data,0,(-1)*$pos);
    else if (($pos = strpos($data,"n\n"))!==false)
        return substr($data,0,(-1)*$pos);
    else if (($pos = strpos($data,"r\r"))!==false)
        return substr($data,0,(-1)*$pos);
    else
        return $data;
}

function http_data($data)
{
    if (($pos = strpos($data,"r\nr\n"))!==false)
        return substr($data,$pos+4);
    else if (($pos = strpos($data,"n\n"))!==false)
        return substr($data,$pos+2);
    else if (($pos = strpos($data,"r\r"))!==false)
        return substr($data,$pos+2);
    else
        return $data;
}

?>
```

9.3.3. xml_dom.inc

```
<?

// 
// xml_dom
//
// helper class for small XML document DOM parsing
// it travels nodes, texts and attributes
// namespace prefixes are prepended to names
// uses UTF-8
//

// dom node types
define('XML_DOM_NODE', 0);
define('XML_DOM_TEXT', 1);
```

Tutorial: News DB Search

```
function xml_esc($str)
{
    return htmlspecialchars($str, ENT_QUOTES, 'utf-8'); // TODO: tomeer citi charseti arii buus?
}

class xml_dom_node
{
    var $xml_name;
    var $xml_type;
    var $xml_level;
    var $xml_index;
    var $xml_parent = NULL;
    var $xml_attr = array();
    var $xml_children = array();
    var $xml_data = array();

    function xml_dom_node($name = "", $type = XML_DOM_NODE, $attributes = array())
    {
        $this->xml_name = $name;
        $this->xml_type = $type;
        if ($type == XML_DOM_NODE && $attributes) $this->xml_attr = $attributes;
    }

    function xml_insert(&$node/*, $index*/)
    {
        $this->xml_children[count($this->xml_children)] = &$node;
        if ($node->xml_type == XML_DOM_TEXT) {
            $this->xml_data[count($this->xml_data)] = &$node->xml_name;
        } else {
            $this->{$node->xml_name}[count($this->{$node->xml_name})] = &$node;
        }
        $node->xml_parent = &$this;
    }

    function xml_remove()
    {

    }

    function xml_dump($beautify = "")
    {
        $str = "";
        $nl = ($beautify ? "\n" : "");
        if ($this->xml_type == XML_DOM_TEXT) {
            $str .= xml_esc($beautify ? str_repeat($beautify, $this->xml_level) . trim($this->xml_name) . "\n" : $this->xml_name);
        } else {
            if ($beautify) $str .= str_repeat($beautify, $this->xml_level);
            $str .= "<{$this->xml_name}" . $nl;
            foreach($this->xml_attr as $attr => $val) $str .= " {$attr}=\"$val" . xml_esc($val) . $nl;
            $str .= ">" . $nl;
            foreach($this->xml_children as $child) $str .= $child->xml_dump($beautify);
            if ($beautify) $str .= str_repeat($beautify, $this->xml_level);
            $str .= "</{$this->xml_name}>" . $nl;
        }
        return $str;
    }
}
```

Tutorial: News DB Search

```
}

// satur visu kopaa
class xml_dom
{
    var $xml_root; // sakne
    var $xml_all; // visas dokumenta nodes

    var $xml_node = NULL; // reference uz parseejamo
    var $xml_level = 0;
    var $xml_index = 0;

    function xml_start_element_handler($parser, $name, $attributes)
    {
        $tmp = &new xml_dom_node($name, XML_DOM_NODE, $attributes);
        if ($this->xml_node) $this->xml_node->xml_insert($tmp); // citaadi taa buus sakne
        $tmp->xml_level = $this->xml_level++;
        $tmp->xml_index = $this->xml_index++;
        $this->xml_all[$tmp->xml_index] = &$tmp;
        $this->xml_node = &$tmp;
    }

    function xml_end_element_handler($parser, $name)
    {
        $this->xml_node = &$this->xml_node->xml_parent;
        unset($this->xml_node->xml_children[count($this->xml_node->xml_children) - 1]->xml_parent);
        $this->xml_level--;
    }

    function xml_character_data_handler($parser, $cdata)
    {
        if (count($this->xml_node->xml_children)) {
            $tmp = &$this->xml_node->xml_children[count($this->xml_node->xml_children) - 1];
            if ($tmp->xml_type == XML_DOM_TEXT) { $tmp->xml_name .= $cdata; return; }
        }
        $tmp = &new xml_dom_node($cdata, XML_DOM_TEXT);
        $this->xml_node->xml_insert($tmp);
        unset($tmp->xml_parent);
        $tmp->xml_level = $this->xml_level;
        $tmp->xml_index = $this->xml_index++;
        $this->xml_all[$tmp->index] = &$tmp;
    }

    function xml_dom($xml)
    {
        $parser = xml_parser_create('UTF-8');
        xml_parser_set_option($parser, XML_OPTION_CASE_FOLDING, FALSE);
        xml_set_element_handler($parser, array(&$this, 'xml_start_element_handler'),
array(&$this, 'xml_end_element_handler'));
        xml_set_character_data_handler($parser, array(&$this, 'xml_character_data_handler'));
        $ok = xml_parse($parser, $xml, TRUE);
        xml_parser_free($parser);
        if (!$ok) return $this = FALSE;
        $this->xml_root = &$this->xml_all[0];
    }
}
```

```
    $this->{$this->xml_root->xml_name}[0] = &$this->xml_all[0];  
}  
  
function xml_eval_xpath($xpath)  
{  
}  
  
function xml_dump($beautify = "")  
{  
    $str = $this->xml_root->xml_dump($beautify);  
    return $str;  
}  
  
function xml_free()  
{  
    for ($i = 0; $i < count($this->xml_all); $i++) unset($this->xml_all[$i]->xml_parent);  
}  
}  
?>
```

9.3.4. lib_smart.inc

```
<?php  
  
function smart_explode($separator, $string, $enclose = "'\"", $escape = "\\", $limit = 0)  
{  
    $inner = false;  
    $positions = array();  
    $strlen = strlen($string);  
    $seplen = strlen($separator);  
    for ($i=0;$i<$strlen;$i++)  
    {  
        if (!$inner && substr($string,$i,$seplen)==$separator) // ir atdaliitaajs  
        {  
            //echo "cut!\n";  
            $positions[] = $i;  
            $i += $seplen-1;  
        }  
        elseif (!$inner && strpos($enclose,substr($string,$i,1))!==false && ($i==0 ||  
strpos($escape,substr($string,$i-1,1))===false)) // saakas iesle  
egums  
        {  
            //echo "to inner!\n";  
            $inner = true;  
        }  
        elseif ($inner && strpos($enclose,substr($string,$i,1))!==false && ($i==0 ||  
strpos($escape,substr($string,$i-1,1))===false)) // beidzas iesle  
egums  
        {  
            //echo "to outer!\n";  
            $inner = false;  
        }  
    }  
    $results = array();  
    $lb = 0;  
    for ($i=0;$i<$limit-1 || ($limit==0 && $i<count($positions));$i++)
```

```
{  
    $results[] = substr($string,$lb,$positions[$i]-$lb);  
    $lb = $positions[$i]+$seplen;  
}  
$results[] = substr($string,$lb);  
return $results;  
}  
?>
```

9.3.5. lib_obcmd.inc

```
<?php  
  
function obcmd_init()  
{  
    if (!isset($GLOBALS["OBCMD_INIT"]) || !$GLOBALS["OBCMD_INIT"])  
    {  
        ob_end_flush();  
        $GLOBALS["OBCMD_INIT"] = 1;  
    }  
}  
  
function obcmd_flush()  
{  
    @ob_flush();  
}  
  
function obcmd_print($text)  
{  
    obcmd_init();  
    echo $text;  
    obcmd_flush();  
}  
?>
```