InteProxy Desktop Users Manual

Status: Juni, 27th 2013 for InteProxy Desktop 1.2.1

Table of Contents

1 General

InteProxy Desktop is a helper for those applications that use HTTP for arbitrary services that also need secure transmission via SSL and user authentication, but are not able to do it by themselves.

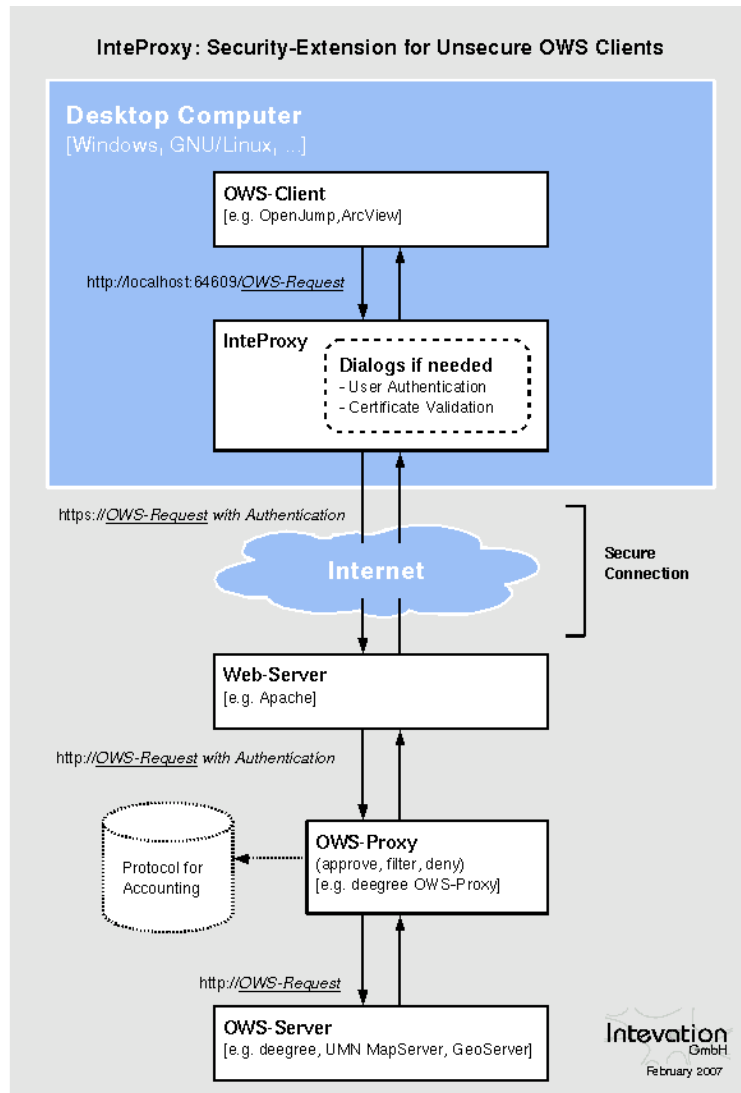*Figure 1: Illustration how InteProxy works*

The proxy runs on the same system as the actual application. InteProxy Server is able to run inside a network and offers secure access to OWS-services for GIS-clients inside the whole network.

A typical case are the services that have been specified by the Open Geospatial Consortium (WMS, WFS, WFS-T, etc.).

Figure 1 illustrates how InteProxy Desktop basically works.

1.1 Support methods for secured OWS

The following technologies to secure OWS services are supported by InteProxy Desktop:

- deegree OWS-Proxy: This product ist directly supported.

- HTTP Basic Authentication: Any server using this standard HTTP method in combination with HTTPS for accessing OWS services is supported.

- URL-Parameter „user=" und „password=": In combination with HTTPS, any OWS server is supported that uses exactly these two URL parameters.

2 Download

The current version of InteProxy Desktop can be found under http://inteproxy.wald.intevation.org.

For the Windows operating system you should download the Windows installer package (e.g. „InteProxy-1.2.1.exe").

3 Installation

MS Windows: Execute the setup program of InteProxy Desktop as administrator on your Windows system. There is no need to install any other applications.

GNU/Linux: Currently no installation packages are available for the various GNU/Linux distributions. The only requirement is the availability of Python, the Python library lxml and python-gtk2 (>=2.10.11). Then you only need to run 'InteProxy Desktop.py' inside your InteProxy source directory.

3.1 Detailed installation (MS Windows)

The following steps are necessary to install the setup-package on MS-Windows.
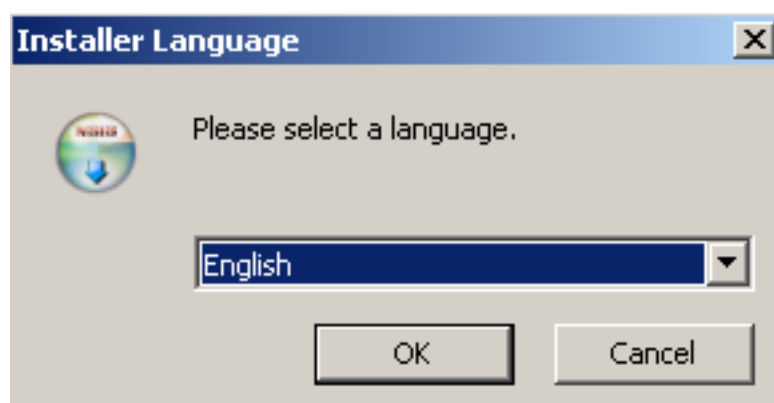
Figure 1: Language selection

After selecting the language the welcome screen will be prompted to you.

Figure 2: Welcome screen

Then you need select the target directory where InteProxy Desktop will be installed to. Please make sure that you have the correct permissions of the folder.
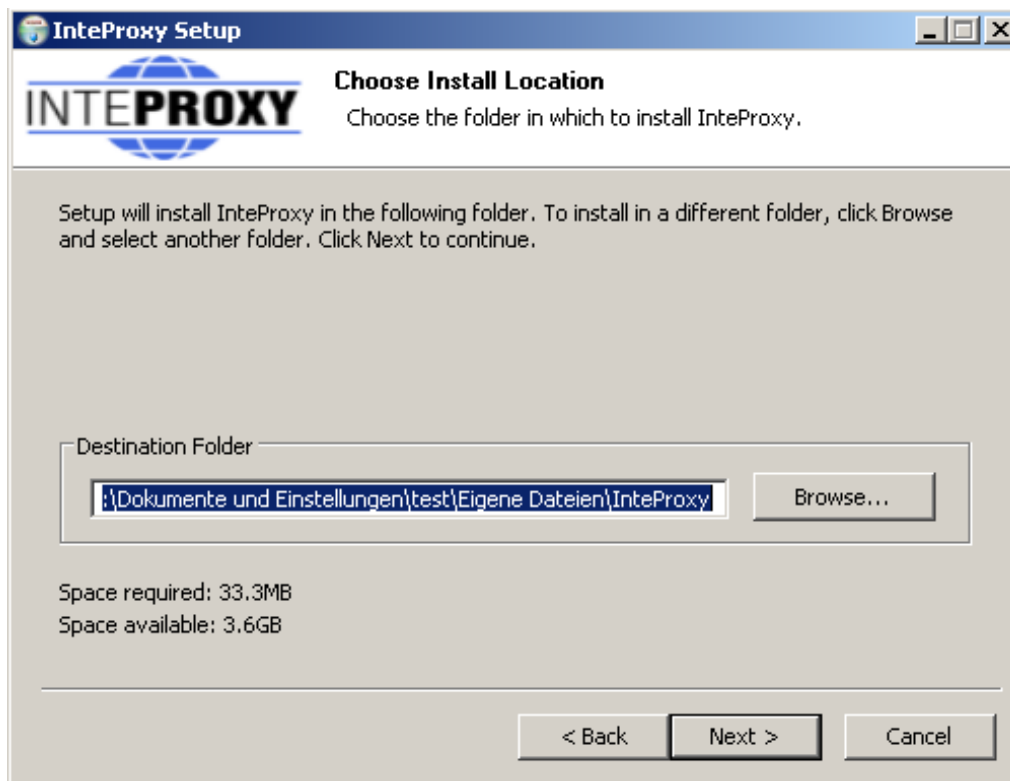
Figure 3: Selection of the installation-directory

If you want a link on your desktop, you can activate the relevant checkboxes on the next step. An entry in the program-menu is set by default.

Figure 4: Set links

Figure 5: Finish the installation

After completing the setup-routine InteProxy Desktop is ready to use.

4   Starting InteProxy Desktop

MS Windows:

Start InteProxy Desktop as user via your Windows menu or with the Desktop-link. A tray-icon inside the task-bar indicates that InteProxy Desktop is running. (see fig. 2).

Figure 2: InteProxy tray-icon embedded in task-bar (GNU/Linux)

Figure 3: InteProxy trayicon embedded in task-bar (Windows XP)

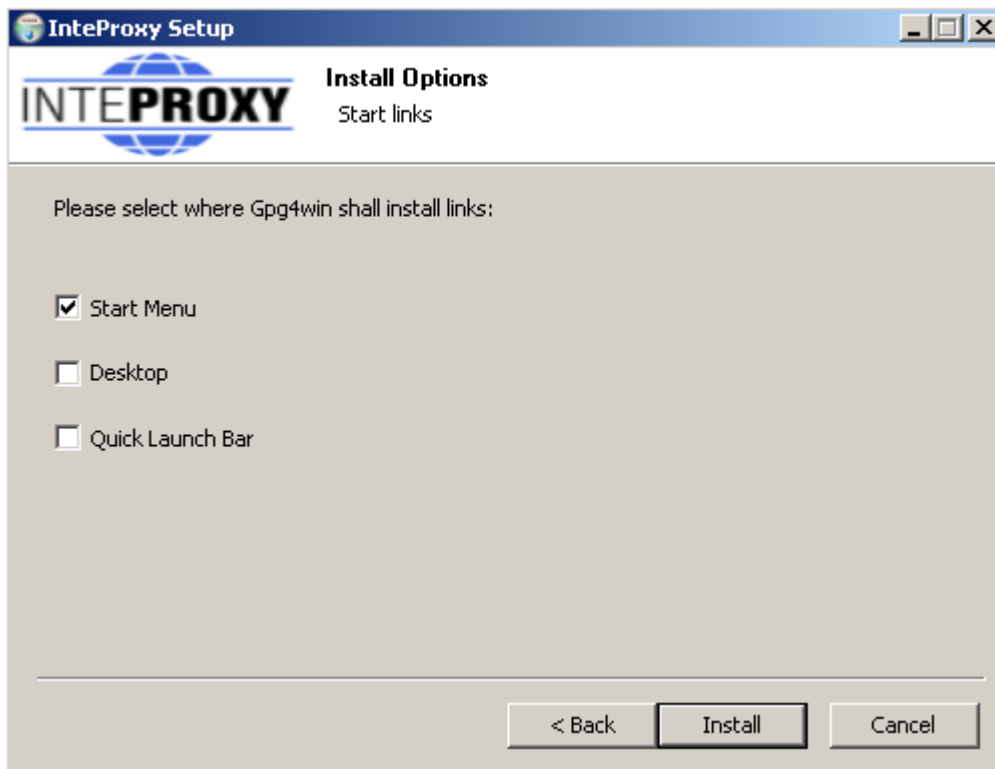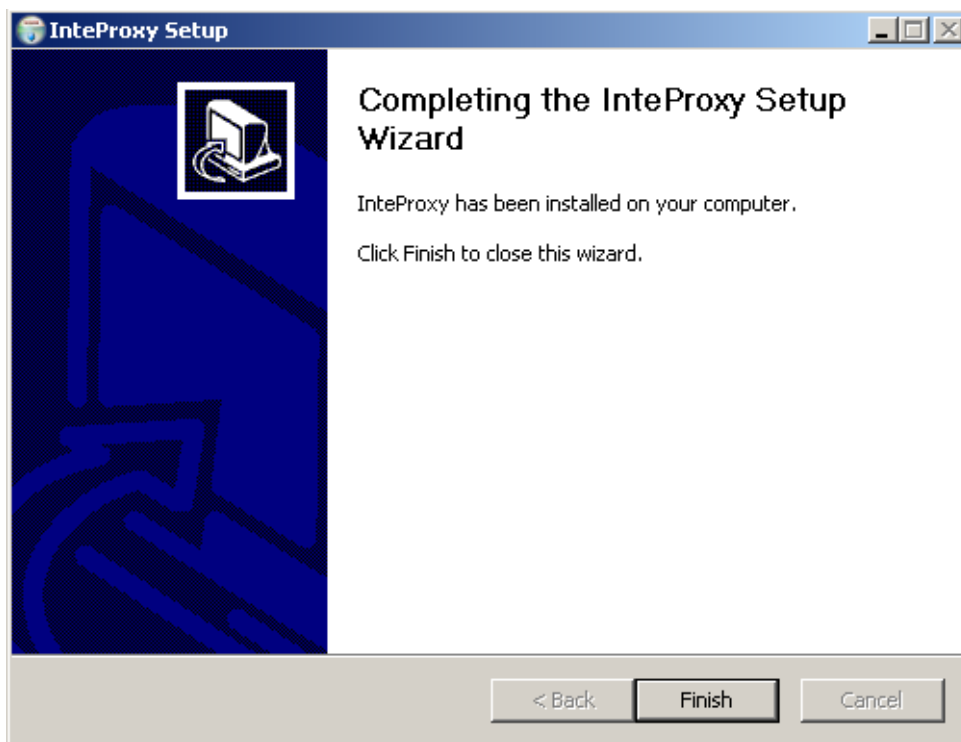The tray-icon presents information about InteProxy Desktop status to the user. Pushing the right mouse-button pops up the GUI of InteProxy Desktop (see fig. 4).

This GUI presents the current InteProxy Desktop-URL; in fig. 4 you can see the default port 64609 listening on the local computer (localhost).

If you like to use InteProxy Desktop as a normal HTTP-Web-Proxy as well, you need to create a configuration file or adopt the file 'inteproxy.cfg' with the needed values. This is the default configuration file InteProxy Desktop looks for, if no dedicated file is provided at startup. If you like to use an alternative file, provide the filename during startup. Change to the folder where InteProxy.exe resides and run the following command:

```
InteProxy.exe --config-file=inteproxy.cfg [optional: more
parameters]
```

Alternatively you can tweak your InteProxy Desktop-entry inside the Windows start menu. More detailed information can be found in section 4.2.
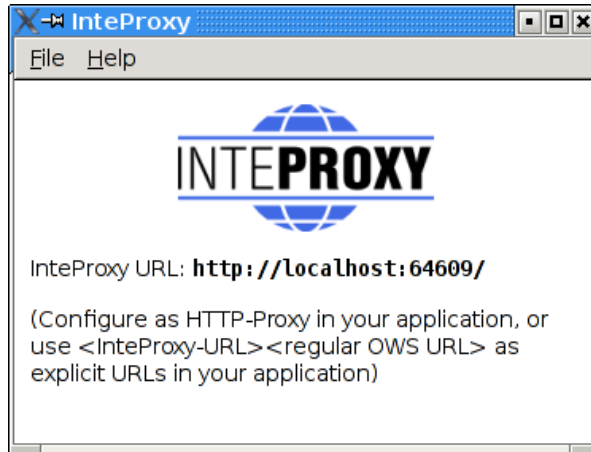
Figure 4: Graphical user interface of InteProxy (GNU/Linux)

GNU/Linux:

In a command shell, change to the directory where you installed InteProxy Desktop and enter this command:

```
$ python InteProxy.py --config-file=inteproxy.cfg [optional:
more parameters]
```

4.1 Calling-Parameters

The available calling parameters of InteProxy.py can be shown with the option -h.

```
usage: InteProxy.py [options]
options:
  -h, --help              show this help message and exit
  --logfile=LOGFILE
  --rewrite-urls
  --port=PORT
  --workers=WORKERS
  --debug-level=DEBUG_LEVEL
  --config-file=CONFIG_FILE
```

The options have the following meanings:

--help	Prints the syntax-help and exits InteProxy Desktop.
--logfile	Writes all output of InteProxy Desktop in the provided logfile.
--rewrite-urls	Translates the URLs inside the server-response to the initialy given host:port-part inside all URLs. This is

	necessary for some OWS-clients (like ArcGIS) to find the correct way to the server through InteProxy Desktop based on the OnlineRessource-URL of the capabilities-document.
--port	Allows the setting of an alternative port. The default is 64609.
--workers	Expert-function: sets the number of parallel worker-threads. The default is 5.
--debug-level	Expert-function: Values larger 0 show the verbosity of output. The larger the value, the more information will be printed out. 0 means no output (default). Currently values of 0, 1 and 2 are implemented.
--config-file	Allows the user to specify alternative configuration files for OWS-services. The default is 'inteproxy.cfg'.

4.2 Configuration-file

The configuration of OWSProxy-services is done with a configuration file. If you provide no OWSProxy-services inside a configuration file, InteProxy Desktop will act as a normal transparent HTTP-web-proxy.

Every source-distribution contains an example configuration file 'inteproxy.cfg'.

If you do not provide any configuration file during starup, InteProxy Desktop looks for the file 'inteproxy.cfg' inside its directory. If this file is not found, InteProxy Desktop acts like a normal HTTP-web-proxy and passes all requests. If you plan to use InteProxy Desktop to secure OGC-services, you need to use a configuration file with appropriate entries.

Furthermore you can configure an upstream proxy with authentification. Therfore you need to uncomment and adopt the sections [inteproxy] and [http_proxy_authentification].

An example configuration file is 'inteproxy.cfg' inside the InteProxy Desktop-sources. This file contains one example of a currently available OWSProxy-service
'inteproxy-demo.intevation.org':

```
# Demo configuration inteproxy.cfg for InteProxy Desktop.
#
#
# This file defines how different remote servers are to be
# accessed. All servers are specified by URLs in the inteproxy–
# rules section.
```

```
#
# For backwards compatibility, an older format is also supported
# where each server is defined by one section for each remote
# server.  How the sections are named doesn't really matter but
# it's a good idea to include the full domain name in the
# section name.  If you want to access to different servers on
# the same host, be careful to give both sections different
# names, by e.g. appending some more information about the
# server.
#
# There might be some additional section with predefined names
# in the uture for other inteproxy settings. The names for those
# section will be chosen so as to not interfere with existing
# section names. Hence the suggestion to use the full domain
# name.
#
# InteProxy-related configuration-entries follow
# !! IMPORTANT:
# !! If you need upstream proxy-authentification, please
# !! uncomment thefollowing entries [inteproxy] and
# !! [http_proxy_authentification] and
# !! adopt them with your credentials!
#
# Authentification against upstream proxy-servers
# [inteproxy]
# https_proxy=http_proxy_authentification
# http_proxy=http_proxy_authentification
#
# Uncomment the following line to suppress an upcoming popup
# that shows fees and access constraints. Allowed values for
# show_terms_dialog option are "false", "no" and "0" to suppress
# the popup, "true", "yes" and "1" to show the popup.
# The default is "true".
#
# show_terms_dialog=false
#
# [http_proxy_authentification]
# host=upstream.proxy.url.com
# port=8080
# username=john
# password=secret

# Main rules section.  The urls setting is a list of urls, one
# on each line.  From the second line on, the lines have to be
# indented.
# Each url is of the form
#     SCHEME://[USER:PASSWORD@]HOSTNAME[:PORT]/PATH
```

```
#
# The credentials (USER and PASSWORD) and the port number (PORT)
# are optional.  If no port is given a default is assumed.
# SCHEME should be one of the supported schemes:
#
#   owsproxy    The remote host is an OWSProxy requiring
#               authentication and https
#
#   basicauth  HTTP Basic Authorization over https
#
# If USERNAME or PASSWORD contain special characters such as
# '%', ':', '@' and non-ascii characters, these should be
# escaped as "%" followed by the two hexadecimal digits of the
# character code.  As a further complication, the percent
# character is not only special in URLs, but also in the syntax
# used in this configuration file where a literal percent
# character has to be quoted by doubling it.  So to encode
# e.g. percent sign, use "%%25" instead of simply "%".
#
# The HOSTNAME and PATH parts of the URL may contain "*"
# characters as wild-cards.  They will match zero or more
# occurrances of any character.  However, a wild-card in the
# hostname will only match in the hostname and a wild-card in
# the path will only match in the path.


[inteproxy-rules]
urls=owsproxy://meier:meier@inteproxy-demo.intevation.org/cgi-
bin/frida-wms
     owsproxy://USERNAME:PASSWORT@mysecure.server.de/mapgate/*
```

5   Examples of usage

InteProxy Desktop offers the possibility to act as an OWSProxy client-program as well as a normal HTTP-web-proxy. The following section shows the two modes with configuration examples in action.

5.1   Userdata

To use the following examples with the demo-services provided in this manual you can use the two groups of userdata.

Server-URL: 'http://inteproxy-demo.intevation.org/cgi-bin/frida-wms'

Two usergroups are available:

- User/Password = meier/meier: can view all WMS layers
- User/Password = schmidt/schmidt: can view all WMS layers except the

roads

5.1 Using InteProxy Desktop as On-demand-proxy

Now in your application prefix the actual URL of the WMS service with the text

„http://localhost:64609/".

This explains the term 'on-demand-proxy'. Whenever you need InteProxy Desktop to be an OWSProxy-client, add the term into your OGC-query. If you do not want to add this term every time, read on at section 5.2 about InteProxy Desktop as HTTP-web-proxy.

InteProxy Desktop caches the authorization data for each URL. Thus you have to authorize only once per URL. Once you stop InteProxy Desktop, all cached data are lost. It is necessary to stop InteProxy Desktop and restart it in case you want to log in for a URL where you already authenticated with another user account.

5.1.1 Simple Test (via Web Browser)

With this test you can check whether InteProxy Desktop basically works and is not blocked e.g. by a firewall.

To test whether InteProxy Desktop runs properly, first make sure InteProxy Desktop is running. Then open the web browser and enter the following URL:

http://localhost:64609/inteproxy-demo.intevation.org/cgi-bin/frida-wms?Request=GetMap&Version=1.1.1&service=WMS&layers=strassenall,sehenswuerdigkeiten&format=image/png&width=200&height=200&srs=epsg:31467&bbox=3427000,5787590,3444000,5800880

The browser will first request at port 64609 on your local system where InteProxy Desktop is waiting for requests. InteProxy Desktop raises a authentication dialog (see fig. 5) for username and password (see section 5.1 for details), if no credentials are stored inside the configuration.

After you finish authentication, your browser should now show a map of the city of Osnabrück, because InteProxy Desktop executed the actual WMS request (frida.intevation.org/cgi-bin/frida_wms?VERSION=1.1&SERVICE=WMS&REQUEST=GetMap...) and relayed the answer to the browser. Especially, InteProxy Desktop uses the protocol „https" and not „http".

5.1.2 Integration in a Desktop-GIS (OpenJUMP)

The desktop GIS „OpenJUMP" (www.openjump.org) is Free Software and supports WMS 1.1.1 layers in the map view.

Start InteProxy Desktop and OpenJUMP. In OpenJUMP choose menu „layers" and the item „add WMS request". You will be asked for a URL. Enter this one:

http://localhost:64609/inteproxy-demo.intevation.org/cgi-bin/frida-wms

For the use of the data two accounts are available (see section 5.1):
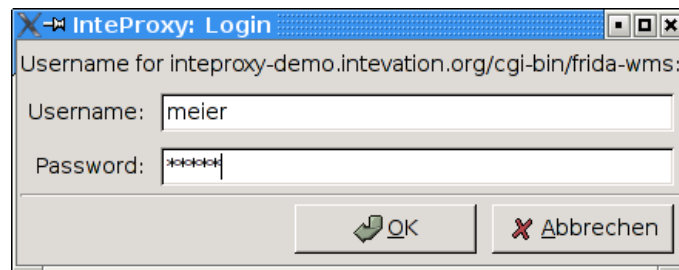
Figure 5: Login-dialogue for authentication (GNU/Linux)

If you authenticate as user „meier“, the layer choice contains more layers than for user „schmidt“. Note, that due to the password caching you need to stop and restart InteProxy Desktop to change the user.

Warning: Please remember that the given example configuration in section 4.2 already contains the user 'meier', so no password dialog will be shown. If you like to have that functionality, remove the user 'meier' and its password from the configuration file and restart InteProxy Desktop.

5.2 Using of InteProxy Desktop as web-proxy

The usage of InteProxy Desktop as normal HTTP-web-proxy has the big advantage, that you do not need to add the clause 'http://localhost:64609' to every OWSProxy-rquest as described in the former chapter 5.1. A HTTP-web-proxy usually gets set system-wide and/or configured for specific applications and then deals with all HTTP-requests.

Based on the entries in the configuration file InteProxy Desktop decides if the requested server is an OWSProxy-secured service or not. If so, it adds username and password automatically.

If you need an internal proxy for you internet connection, you need to publish this to InteProxy Desktop as well. If your Intranet-proxy requires authentification, uncomment the relevant sections in your configuration and adopt them accordingly (see section 4.2).

5.2.1 Configuration as web-proxy under GNU/Linux

If you already have configured an internal web-proxy to connect to the Internet you need to tell InteProxy Desktop that this is the proxy to handle over all requests not dealing with secured SDI-services.

In this case you need to export the URL of your external web-proxy as an environment variable in the same shell where you will start InteProxy Desktop afterwards:

```
$ export http_proxy="http://myproxyhost.fqdn:port"
$ export https_proxy="http://myproxyhost.fqdn:port"
```

Now you can start InteProxy Desktop locally in this shell; it uses your internal proxy as well.

```
$ python InteProxy.py --config-file=inteproxy.cfg [optional
parameters]
```

Please note, that InteProxy Desktop needs to be added into your application, e.g. your web browser or your desktop-GIS.

Mozilla Firefox: Edit-> Preferences -> advanced -> network-tab -> settings -> manual proxy configuration: Enter localhost as HTTP-proxy and the port InteProxy Desktop is running (default is 64609).

5.2.2 Configuration as web-proxy under Windows

The internal web-proxy for InteProxy Desktop can be adjusted as follows:

Start -> system resources -> system -> extended -> Environment variables

There you can add the environment variables 'http_proxy' and 'https_proxy' (depends on your network configuration) and set appropriate values. An example is shown in fig. 6:
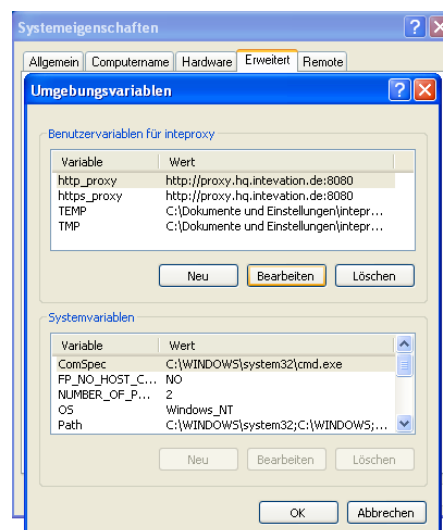

Figure 6: Adjust environment-variables (Windows XP)

After adjusting the proxy-settings for your internal web-proxy you can start InteProxy Desktop with the following parameters:

```
InteProxy Desktop.exe --config-file=inteproxy.cfg [optional
parameters]
```

Naturally you need to tell your Windows-browser which proxy-address to use:

Mozilla Firefox: Edit-> Preferences -> advanced -> network-tab -> settings -> manual proxy configuration: Enter localhost as HTTP-proxy and the port InteProxy Desktop is running (default is 64609).

Internet Explorer: Internet options -> Connections -> LAN-settings -> settings -> Proxy-server -> advanced: add 'localhost' and the port InteProxy Desktop is running (default is 64609).

Please note, that the steps above are translated from the German GUI due to the lack of English windows version.

Warning: You need special administrative rights to adopt the settings described above.

5.2.3   Using of InteProxy Desktop with a Desktop-GIS (example OpenJUMP)

OpenJUMP is a Java application. Java-applications also need to know information about proxy-servers. Therefore you can add the needed parameters to the startup-script of OpenJUMP (usually openjump.bat, openjump-mac.sh or openjump-unix.sh):

```
–DproxySet=true –DproxyHost=localhost –DproxyPort=64609
```

The entries above assume that InteProxy Desktop is running on the local computer with its default port 64609. Additionally the OWSProxy-services are configured properly inside the configuration file.

Now you are able to use OWSProxy-secured services inside OpenJUMP as you are used to. Enter the secure OWSProxy-Service using the menu „layers" and their item „add WMS request". The URL is the following:

http://inteproxy-demo.intevation.org/cgi-bin/frida-wms

InteProxy Desktop now recognizes that you are querying an OWSProxy and asks you for a OWSProxy-user with password which will be used by InteProxy Desktop to authenticate against the OWSProxy.

After connection a window with the servers fees and access-constraints will be shown (see fig. 7).
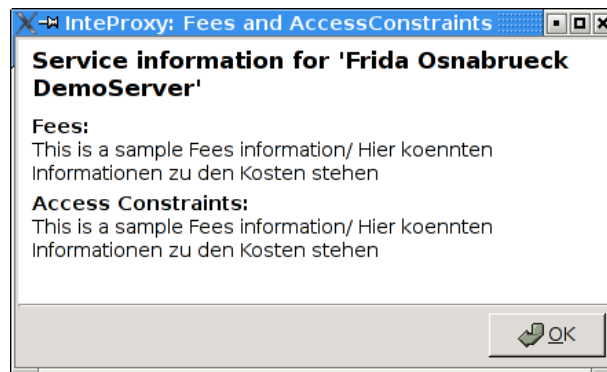
Figure 7: AccessConstraints and fees dialog after loading a OWSProxy-service

This information will be extracted from the WMS-server after loading initially.

5.2.4   Using inside a JavaApplet based webmapping-application (Example Niedersachsenviewer Plus)

Including secured WMS-layers inside a (non-secure) JavaApplet based webmapping-application assumes that the WMS-query gets done by the client-program instead of the server.

The JavaApplet 'Niedersachenviewer Plus' (http://geoportal.geodaten.niedersachsen.de/organizer/) includes the possibility to add own WMS-services. A working Java-installation on the clientside needs to be available. To include secured WMS-layers in a webmapping-client InteProxy Desktop needs to be configured as normal HTTP-web-proxy as described in chapter 5.2.

The tab 'layercontrol' provides a symbol (green plus) to add own WMS-services to the map. Depending on your OWSProxy-user the layers are available in tab 'layercontrol'. The layers can be enabled for viewing by checking the appropriate box.

5.2.5   Using with a pure JavaScript-base webmapping application (example OpenLayers)

OpenLayers (http://www.openlayers.org) is a free, pure JavaScript-base API, which facilitate the creation of dynamic maps in webmapping applications. OpenLayers completely runs inside the client (web browser).

A testinstallation is made available at

http://demo.intevation.de/openlayers/

There you have two simple text boxes where you can specify the secured WMS-services by URL and its layers.

You can use the URl

http://inteproxy-demo.intevation.org/cgi-bin/frida-wms

for testing. Depending on your chosen OWSProxy-user (see sec. 5.1) you can use the layers gewaesser, sehenswuerdigkeiten, strassenall, gruenflaechen for testing.

Note:

Naturally you need a local configuration for your InteProxy Desktop as HTTP-web-proxy (see sec. 5.2) as well as a working browser configuration.

6 Note when using Intranet-Proxies

In case all HTTP/HTTPS request of your system need to go over a proxy service, you might experience various problems depending on the actual configurations. Here are some hints on what you can do to cope with this:

- Configure environment variable „https_proxy“: InteProxy Desktop interprets and uses the environment variable „https_proxy“. In case this variable defines e.g. „http://proxy.hq.sample“, InteProxy Desktop will submit all requests to this service. This holds for both, MS Windows and GNU/Linux. Attention: Problems may occur when concurrently defining „http_proxy“.

7 Note when using secured WFS-services

The configuration of WFS-services is in line with the configuration of WMS-services , described in the chapter 4.2 Configuration-file on page 10. Add the secured WFS-service in the main rules section '[inteproxy-rules]' in the configuration file 'inteproxy.cfg'. An example configuration is provided in the following snippet of the configuration file 'inteproxy.cfg':

```
[inteproxy-rules]
urls=owsproxy://USERNAME:PASSWORT@mysecure.server.de/owsproxy/wrapper
```

7.1 Simple Test (via Web Browser)

With this test you can check whether InteProxy Desktop basically works and is not blocked e.g. by a firewall.

To test whether InteProxy Desktop runs properly, first make sure InteProxy Desktop is running. Then open the web browser and enter the following URL:

http://localhost:64609/mysecure.server.de/owsproxy/wrapper?
Request=GetCapabilities&Service=WFS&Version=1.1.0

The browser will first request at port 64609 on your local system where InteProxy Desktop is waiting for requests. InteProxy Desktop raises a authentication dialog (see fig. 5) for username and password (see section 4.2 for details), if no credentials are stored inside the configuration.

After you finish authentication, your browser should now display a GetCapabilities document.

8   Limitations and Perspectives

An adoption for MapBenders OWSProxy is relatively easy to realize („sid" and „wms" instead of „user" and „password"), but so far practice showed no demand for this.

User-friendly (graphical dialogs) of these (and other) configurations are planned.

InteProxy Desktop does not work with webmapping-clients which parse a WMS-URL on server-side. This applies to NiedersachsenViewer, Mapbender and iGeoporal. In this cases InteProxy Desktop eventually needs to be installed and configured on server-side to pass the secured connections between portal and OWS-services. This step is already prepared, but it will be tested and described in detail in the upcoming version of InteProxy Desktop.

See known bugs and report new problems:

> http://wald.intevation.org/tracker/?group_id=23

9   Version History

Current release is 1.2.1 from June, 27th 2013.

New since version 1.2.1 from June, 27th, 2013:

- Solved issue No. 617. Fees dialog is now displayed when deflate / gzip compression is requested.

New since version 1.2.0 from June, 25th, 2013:

- Deflate/GZIP-compression for response-streaming for xml documents.

New since version 1.1.0 from January, 11th, 2012:

- Improvement: Response-streaming for xml documents.

New since version 1.0.3 from October, 28th, 2010:

- Adjustments for WFS 1.0 and 1.1 requests

New since version 1.0.1 from May, 26th, 2010:

- Correction: Bug #1217 GetCapabilites URL checking. GetCapablities Request are now case insensitive.
- Improvement: Added a 'show_terms_dialog' option in inteproxy.cfg to suppress an upcoming popup that contains information about fees and access constraints.

New since version 1.0.0 from December, 10th, 2009:

- Improvement: the sample configurationfile contains optional upstream proxy-settings.
- Improvement: Upstream-Proxy-Authentification
- Improvement: Package for OpenSuse 10.x
- Improvement: Development and documentation of InteProxy-Server
- Improvement: Configurationscript for InteProxy-Server
- Improvement: better Logging
- Improvement: Added Unit-Tests

New since version 0.4.1 from July 14th, 2008:

- Improvement: No administrator-rights for installation needed
- Improvement: secure WMS-adresses of LGN added to configurationfile
- Improvement: rewrite also HTTPS-URLs
- Correction: Bug #616: keyboard-focus for passwort-input

New since version 0.3.1 from April, 17th 2008:

- parameter --rewrite-urls added
- Parameter --allow-shutdown removed, since it is not needed anymore
- Improvement: HTTP-queries are passed through without touching the HTTP version (HTTP 1.0 and HTTP 1.1 are supported).
- Improvement: more tests for automated testing
- Improvement: Windows-Installer ported to NSIS-technology

New since version 0.3.0 from February, 27th 2008:

- Correction: For the web-proxy method, InteProxy will not any more connect

HTTP via the HTTPS proxy. This lead to problems for some intranet proxy configurations and resulted e.g. in problems using the web browser.

New since version 0.2.0 from May, 18th 2007:

- New: Support of „Basic Authentication" of the HTTP Protocol. Now InteProxy can access servers that use this standard methode to secure OWS services.
- Improvement: In some cases problems occurred combining ArcGIS as client and deegree OWS-Proxy as server. This protocol error is now neutralized by InteProxy, although the actual problem lies somewhere else.
- Improvement: In some cases, when InteProxy works as HTTP-Proxy, errors occurred e.g. for "/favicon.ico". This has been fixed.
- Solved issue No. 243 „Authentication dialog not always in foreground", http://wald.intevation.org/tracker/index.php?func=detail&aid=243&group_id=23&atid=178
- demo.cfg has been extended with some explanations and hints to better prevent wrong configurations.
- Improvement: Access Constraints: Are these data not sent in correct format, now InteProxy recognizes this case and doesn't react wrongly any more.

New since version 0.1.2 from March, 29th 2007:

- New: GUI, plattform independent (GTK2)
- New: HTTP-WebProxy and introduction of a configuration file
- New: Internationalisation (I18N) (currently only german available)
- New: Access-Constraints and Fees provided in dialogs
- Internal code-restructuring
- Solved issue No. 369 "InteProxy does not handle redirects correct", http://wald.intevation.org/tracker/index.php?func=detail&aid=369&group_id=23&atid=178

New since version 0.1.1 as of December 4th 2006:

- New: Command-line parameter „--debug-level" to switch on debug information. This is an expert function. Values larger than 0 describe the level of details for the debug information (currently only one level is implemented). A value of 0 means that no debug information are collected. The default is 0.
- New: Command-line parameter „--worker" to define the number of parallel connections. The default is 5.
- Solved issue No. 271 „Firefox/Konqueror couldn't display Image

(REQUEST=GetMap).", http://wald.intevation.org/tracker/index.php?func=detail&aid=271&group_id=23&atid=178

- New: Support of WFS für deegree OWS-Proxy.

- Improved manual.

- calling-parameters added

- Szenario for usage of InteProxy as webproxy AND OWSProxy-client with OpenJUMP