



WafL User Guide

Version 0.5.3.17

The information contained in this publication does not include product warranties, and any statements provided in this manual should not be interpreted as such.

All versions of software and documentation are in the development phase. Please contribute the development by reporting bugs and sending suggestions to authors, using email address: `waf1 (at) matf.bg.ac.yu`.

Contents

1. WafI Distribution	3
1.1. Available Software	3
1.2. WafI Package Contents	3
1.3. Windows Installer Package	4
1.4. Linux Package	4
2. Setup and Configuration	5
2.1. Command Line Interpreter	5
2.2. WafI Module for Apache Web Server	8
2.3. WafI Module for MS IIS	15
2.4. Standalone WafI Web Server	19
2.5. Configuring WafI Web Service	20
2.6. Database Connections	23
2.7. Logs	26
2.8. Development Tools	27
Index	28

1. Wafl Distribution

1.1. Available Software

The following versions of Wafl interpreter are available:

- Command Line Interpreter (Windows)
- Command Line Interpreter (Linux)
- Web Server Module for MS IIS (MS Windows)
- Web Server Module for Apache (MS Windows)
- Web Server Module for Apache (Linux)
- Standalone Wafl Web Server (MS Windows)

Windows software require Win32 platform. It is tested with MS Windows XP, MS Windows 2003 Server, MS Windows 2000 and MS Windows 98.

Linux software require unixODBC library for database access.

1.1.1. Database Drivers

Wafl connects to databases using provided drivers . The following drivers are available:

Driver	Notes
ODBC	For DB connections using ODBC. Windows version uses Windows ODBC drivers. Linux version uses unixODBC library.
DB2	Provides full native connections to IBM DB2 databases. Each major version of DB2 should use specific software build. Supports ODBC connections to other databases, to, but some compatibility issues may arise. Different versions of DB2 are supported, using specific drivers DB2v5 (for versions 5 to 7), DB2v8 (for versions 8 and 8.1) and DB2v82 (for 8.2 and later).

Table 1: Database drivers

1.2. Wafl Package Contents

Setup package contains files in the following subfolders:

- `Bin` folder contains WafL binaries, including WafL DB drivers;
- `Doc` folder contains user manuals and other documents, including WafL tutorials;
- `Lib` folder WafL libraries;
- `Log` folder is the default location for WafL services log files;
- `Resources` folder contains WafL icons and WafL file templates;
- `Support` folder contains some additional files.

1.3. Windows Installer Package

WafL for Windows is distributed using installer packages. Installers do the following:

- creates WafL program files folder and appropriate subfolders;
- copies WafL files;
- creates icons in Start Menu;
- sets environment variables `PATH` and `WAFL_PATH`;
- registers WafL file types;
- registers WafL uninstaller.

WafL Services are configured using manual procedure described below.

1.4. Linux Package

WafL for Linux is distributed using archived files.

No automatic installer is provided, yet.

2. Setup and Configuration

2.1. Command Line Interpreter

WafL Command Line Interpreter `clWafL` is designed to interpret WafL programs in operating system command line environment.

Program Setup

Program `clWafL` requires no special setup. It is enough to have the program in the current working directory or in program path environment.

Libraries Setup

If WafL programs use WafL libraries that are neither located in the same directory as the programs, nor the full library file path is specified, the interpreter tries to find the library files in `lib` subfolder of WafL base folder (the value of `WAFL_PATH` environment variable).

2.1.1. Manual Setup Procedure for Windows

1. Create the base folder for WafL files (we will refer the folder by `[WafLRoot]`), e.g.
`C:\Program Files\WafL`;
2. Copy the full content of `Windows` setup subfolder to `[WafLRoot]`;
3. Add `[WafLRoot]\Bin` to environment `PATH` variable;
4. Add `WAFL_PATH` environment variable with value `[WafLRoot]`;
5. Check if the users of the system have the rights to execute WafL binaries and read libraries and other files.

2.1.2. Setup Procedure for Linux

Supported Linux Versions

Command line interpreter should run on virtually any Linux OS for Intel platform. It is tested with Red Hat 7.3 and Red Hat 9.0.

Procedure

1. Create the base folder for Wafl files (we will refer the folder by [WaflRoot]), e.g.
/usr/local/wafl;
2. Copy the full content of Linux setup subfolder to [WaflRoot];
3. In the system binaries directory (probably /usr/bin) make a link to [WaflRoot]/bin/clWafl. For example:

```
link /usr/local/wafl/bin/clwafl /usr/bin/clwafl
```

4. Add [WaflRoot]\Bin to environment program path;
5. Add WAFL_PATH environment variable with value [WaflRoot];
6. Check if the users of the system have the rights to execute Wafl binaries and read libraries and other files.

2.1.3. Using Command Line Interpreter

Command line interpreter is used in the following form:

```
clWafl [<options>] <wafl_program>
```

where <wafl_program> may be:

```
<wafl_program_file_name> [<arguments>]
```

or

```
-code <wafl_source_code> [-args <arguments>]
```

The following options are supported:

Option	Usage
-env:<name>=<value>	Set the environment variable <name> to <value>.
-libdir:<lib_directory>	Set the library search directory to <lib_directory>.
-args <arguments>	The Wafl program command line arguments follow from this point to the end of the command line.
-listlib	Report the content of the Wafl system library.
-wait	Wait for a key after the end of the evaluation. If the interpreter is used in a graphical environment (by double click, or drag & drop), this option lets the command line window open after the evaluation, to provide the user with the time to record the result.
-repeat:<n>	Repeat the program evaluation for <n> times. Used primarily for debugging and performance measurements.
-memory	Reports the memory manager status.

Option	Usage
	Used primarily for debugging and performance measurements.
-timer	Measure the program evaluation duration. Used primarily for debugging and performance measurements.
-source	Output the source code. Used primarily for debugging and testing.
-msgs	Output all interpreter messages, even if the evaluation is successful. Used primarily for debugging and testing.
-dbdriver:<driver-name> -db:<database-name> -user:<username> -pwd:<password>	These options set database connection parameters. They correspond (in the specified order) to service parameters dbDriver, dbAlias, dbUser and dbPassword. See 2.6 <i>Database Connections</i> , at page 23.

Table 2: Command line interpreter options

2.1.4. Limitations

Command line interpreters do not contain some Web specific features.

- Command line environments can not provide Web sessions and specific Web request parameters. Thus, the following functions are not supported in command line versions:
 - Form and FormValue;
 - Session and SessionValue;
 - Service and ServiceValue;
 - httpHost, httpScript and httpPath.
- Transactions in command line environment do not support commands for setting the session parameters:
 - SET;
 - RESET.
- Current versions of command line interpreters do not support Web dialogs. The following functions are not supported:
 - ask – not available;
 - answerAction – not available.

2.1.5. Specifics

Command line interpreters include some command line specific features:

- Single function for user interaction:
 - `input - (String -> String)` outputs the message, waits for user input and returns the input as a result;
- Program command line arguments are available by:
 - `cmdLineArgs - (-> List[String])` returns the command line arguments list, having the program file name as a first element.
- There is a function that echoes a string value to standard output and continues the evaluation:
 - `cmdPrint - (String -> String)` is designed for usage in database scripting and shell scripting.
- Shell commands execution is supported by:
 - `cmdShellExecute - (String -> String)`
Function executes the given command and returns the output text as a result.
- Last shell command status code is returned by:
 - `cmdLastError - (-> int).`

While `cmdPrint` function is quite simple, it allows for rather complex usage. For example, `echoFn(x, f)` outputs `f(x)` and returns `x`:

```
'test'
->echoFn( \s: 'Message: ' + s + '\n' )
->strUpperCase()
where{
  echoFn(x, f) =
    {# x, f(x)->cmdPrint() #}.1;
}
```

2.2. WafL Module for Apache Web Server

The setup of WafL module for Apache Web server includes the placement of files in appropriate folders and configuration of WafL Web services.

2.2.1. Setup Procedure for Linux

Supported Linux Versions

WafL Apache module should run on virtually any contemporary Linux OS for Intel platform. It is tested with Red Hat 7.3 and Red Hat 9.0.

Supported Apache Versions

WafL Apache module is developed for Apache 2.0. It is tested with version 2.0.48 and later. The specific module design require the execution in single process with multiple threads.

Procedure

First steps are the same as described in section 0

Setup Procedure for Linux, at page 5.

After that, it is necessary to configure Apache Web server, WafI module and WafI services. All configuration steps are based on editing the main Apache configuration file `httpd.conf` (probably `/etc/httpd/conf/httpd.conf`).

1. *Configuring Apache to run in single process with many threads.*

Apache 2 supports running in three modes. The mode is configured during the Apache compilation. Each of the modes requires specific configuration:

a. Multiple processes, with single thread per process:

- Changes are located in section:

```
<IfModule prefork.c>
...
</IfModule>
```

- Limit the number of processes to 1:

```
MaxClients 1
```

- Disable automatic process replacing:

```
MaxRequestsPerChild 0
```

- NOTE: Because of limiting the Web server to single process with single thread, this mode is inefficient when compared to other modes. Thus, please avoid running WafI in this mode, even if it is the default mode for certain versions of Apache on Linux.

b. Multiple processes, with multiple threads per process:

- Changes are located in section:

```
<IfModule worker.c>
...
</IfModule>
```

- Limit the number of processes to 1:

```
MaxClients 1
```

- Disable automatic process replacing:

```
MaxRequestsPerChild 0
```

- Set higher number of allowed threads per process:

```
ThreadsPerChild 50
```

- NOTE: The specified count of threads per process is not changed during the server execution. It is necessary to accommodate the count to specific conditions

and other server duties, considering the limitations of certain Linux editions related to processes and threads.

- c. Constant number of processes with variable number of threads per process:

- Changes are located in section:

```
<IfModule perchild.c>
...
</IfModule>
```

- Limit the number of processes to 1:

```
NumServers 1
```

- Disable automatic process replacing:

```
MaxRequestsPerChild 0
```

- Set higher number of allowed threads per process:

```
MaxThreadsPerChild 50
```

- NOTE: From the point of Waf module implementation, this execution mode is the most appropriate one.

2. In the same configuration file declare the Waf module loading:

```
LoadModule waf_module [WafRoot]/bin/mod_waf.so
```

(replace [WafRoot] with full path to Waf install directory).

3. Register Waf programs extensions either at server level (out of all sections) or for specific directories:

```
AddHandler waf_module .waf
AddHandler waf_module .wfl
AddHandler waf_module .wsvc
```

4. Finally, it is necessary to configure Waf services. A same Apache Web server can host many Waf services. A Waf service is configured in a section corresponding to its base directory:

```
<Location "[path]">
...
</Location>
```

where [path] is corresponding Web (virtual) path to resources of the service (e.g. "/" or "/cgi-bin"). The service parameters are specified in that section in the following form:

```
[directive] [parameter]
```

5. Optionally, to avoid errors in processing of configuration files if Waf module is disabled, one may fence the described configurations by a condition:

```
<IfModule mod_waf1.c>
...
</IfModule>
```

The detailed description of Waf1 service parameters for Apache Web server follows in section 2.2.3 *Configuring Service*, at page 12.

2.2.2. Setup Procedure for Windows

Supported Windows Versions

Waf1 module for Apache Web server is designed for Win32 platform. It is tested with MS Windows XP, MS Windows 2003 Server, MS Windows 2000 and MS Windows 98.

Supported Apache Versions

Waf1 Apache module is developed for Apache 2.0. It is tested with version 2.0.48 and later. The specific module design require the execution in single process with multiple threads. That is the default Apache execution mode for Windows.

Procedure

First steps are the same as described in section 2.1.1 *Manual Setup Procedure for Windows*, at page 5.

After that, it is necessary to configure Apache Web server, Waf1 module and Waf1 services. Apache configuration file `httpd.conf` is located in `conf` subfolder of Apache installation. Configuration procedure is similar to the procedure for Linux, as described in section 2.2.1 *Setup Procedure for Linux*, at page 8.

The most significant difference is in configuring the Apache execution mode:

1. Apache for Windows has an additional module `prefork` for process and thread management. It is configured in the following section:

```
<IfModule prefork.c>
...
</IfModule>
```

In *prefork* execution mode it is assumed that there is a single process with multiple threads:

- Set the total number of threads per process:

```
ThreadsPerChild      250
```

- Disable automatic process replacing:

```
MaxRequestsPerChild  0
```

2. In the same configuration file declare the Waf1 module loading:

```
LoadModule waf1_module [Waf1Root]/bin/mod_waf1.so
```

(replace `[Waf1Root]` with full path to Waf1 install directory).

The other configuration steps are same as for Linux. The detailed description of WafI service parameters for Apache Web server follows in section 2.2.3 *Configuring Service*, at page 12.

NOTE: In Apache configuration file for Windows symbol '/' is used in paths, instead of '\\'.

2.2.3. *Configuring Services*

The configuration of each particular service can be specified either in Apache configuration file or in a separate service configuration file. In either case it is necessary to specify some options in Apache configuration file `httpd.conf`, but it is highly recommended to use separate service configuration files for details of WafI service configuration.

Configuration file may contain some confidential information, like username and password for database access. Thus, it is recommended to carefully define access rights on configuration files.

Defining a New Virtual Directory or a New Web Location

A new Web location or a new virtual directory is defined in a same way as without WafI module. However, it is necessary to specify how WafI programs should execute in each particular Web location or virtual directory in which WafI programs may take place. New options are added in appropriate `Location` section.

The only mandatory parameter is the WafI service name. It is defined as:

```
WafIService      Amazon
```

Other parameters are optional and their appearance depends on a chosen form of WafI configuration.

Here is a simple configuration example:

```
<Location "/amazon">
  WafIService      Amazon
  ...
</Location>
```

It is highly recommended to set `WAFI_PATH` environment variable for each WafI service, to allow for easier location of WafI database drivers.

Configuring Services Using Separate Service Configuration Files

WafI service configuration files use XML syntax. Recommended configuration file name is `www.wsvc`, and recommended configuration file location is the base service directory.

The location and name of service configuration files are specified in Apache configuration file as a value of `ConfigFile` parameter:

```
ConfigFile      <pun naziv datoteke>
```

The path is specified following the rules for specific operating system.

A Windows example:

```
<Location "/amazon">
  WafIService      Amazon
  ConfigFile      "D:\WebApps\Amazon\wwwroot\www.wsvc"
```

```
</Location>
```

A Linux example:

```
<Location "/amazon">
    WafIService      Amazon
    ConfigFile       "/webapps/amazon/wwwroot/www.wsvc"
</Location>
```

The elements of service configuration files are described in section 2.5 *Configuring WafI Web Service*, at page 20. An example of service configuration file is available in `Doc` subfolder of WafI installation.

Configuring Services Using Apache Configuration File

If a service is completely configured using Apache configuration file `httpd.conf`, then `ConfigFile` parameter must not be used. The following directives are used to set service configuration:

Directive name	Description
<code>WafIService</code>	Unique service name. Mandatory.
<code>Host</code>	The base path to service Web location. It is used to distinguish different virtual servers. It consists of host name and the path specified in the location section header. Even if the location is used locally, it is necessary to use the host address specified here. Mandatory.
<code>LibDirectory</code>	Directory with WafI library files. It is used to identify the location of common WafI libraries. Optional. Mandatory if common libraries are used without full path specification.
<code>DefaultContentType</code>	Default type of MIME resources evaluated by WafI programs. It is used to create the appropriate HTTP header, if program result is not of <code>MimeResource</code> type. Optional. Default value is <code>"text/html"</code> .
<code>MaxThreads</code>	The maximal number of WafI evaluator threads. It is recommended to set this parameter slightly above the processor count: <div style="text-align: center;"> 1 processor 4 threads 2 processors 6 threads 4 processors 8 threads </div> Optional. Default value is 4.
<code>SetWafIEnv</code>	Sets the value of WafI execution environment variable. It has two arguments: the first is a variable name and the second is a variable value. Optional. Can be used many times, thus defining many environment parameters.
<code>SetWafIParam</code>	Sets the value of WafI service parameter, that WafI programs

Directive name	Description
	<p>access by functions <code>Service</code> and <code>ServiceValue</code>.</p> <p>It has two arguments: the first is a parameter name and the second is a parameter value.</p> <p>Optional. Can be used many times, thus defining many service parameters.</p> <p>It is highly recommended to set <code>WAFL_PATH</code> environment variable for each WafI service, to allow for easier location of WafI database drivers.</p>

Table 3: Apache service configuration directives

An example of service configuration for Linux:

```

<Location "/cgi-bin">
    WafIService      Svc1
    Host              127.0.0.1/cgi-bin
    LibDirectory      "/usr/local/wafI/lib/"
    DefaultContentType text/html
    MaxThreads        4
    SetWafIEnv        WAFL_PATH      /usr/local/wafI
    SetWafIEnv        DB2CODEPAGE    1250
    SetWafIEnv        DB2INSTANCE    db2inst1
    SetWafIParam      defaultUserId  17
</Location>

```

The same example for Windows:

```

<Location "/cgi-bin">
    WafIService      Svc1
    Host              127.0.0.1/cgi-bin
    LibDirectory      "c:\Program Files\WafI\Lib\"
    DefaultContentType text/html
    MaxThreads        4
    SetWafIEnv        DB2CODEPAGE    1250
    SetWafIParam      defaultUserId  17
</Location>

```

2.2.4. Configuring Database Connections

Database connection configuration is a part of service configuration. Any service can connect to a single database. WafI module pools database connections.

If service is configured in Apache configuration file, database connection is configured using directives: `DbDriver`, `DbAlias`, `DbUser` and `DbPassword`. They correspond to parameters described in section 2.6 *Database Connections*, at page 23.

Database Drivers

The only database driver for Linux is ODBC, using `unixODBC` library. Windows database drivers include ODBC and DB2.

Using DB2 Database Driver With Apache

The native DB2 driver is available only for Windows. To access DB2 databases using native DB2 drivers, it is necessary to backup the original file `ApacheWA.dll` and to rename `ApacheWaDb2.dll` to `ApacheWA.dll`.

2.3. WafI Module for MS IIS

The setup of WafI module for MS IIS includes the placement of files in appropriate folders and configuration of WafI Web services.

2.3.1. WafI Module Setup Procedure

Supported Windows Versions

WafI module for MS IIS is designed for Win32 platform. It is tested with MS Windows XP, MS Windows 2003 Server and MS Windows 2000.

Supported MS IIS Versions

WafI module for MS IIS is designed and developed for versions 5 and 6. The most of functionality is available on IIS version 4, but with some possible issues.

Procedure

The software installation procedure is described in section 2.1.1 *Manual Setup Procedure for Windows*, at page 5.

NOTE: Because of an unpleasant IIS bug, the path to WafI binary files must not contain blank spaces. If WafI is installed in Program Files folder, please copy binaries to another location, with no blank spaces in folder path, to avoid the problems.

The following steps describe how MS IIS is configured to use WafI module for WafI programs processing¹:

1. Create the base folder for new Web service;
2. Start Internet Services Manager (*Start/Settings/Control Panel/Administrative Tools/Internet Services Manager*);
3. Create a new *Web Site* for the service and set its base folder (from step 1), or, if service is a part of existing Web site, create a new virtual directory. In both cases it is necessary to set *Read* and *Run Scripts* rights. During the development it is often useful to allow directory browsing, too;
4. Open the Web site configuration dialog (right click the site, and select *Properties*) and find page *Home Directory*;
5. Allow execution of scripts by selecting *Scripts Only* under *Execute Permissions*;
6. Select appropriate service isolation level (*Application Protection* option). If many different WafI services access DB2 databases with different code pages using the native driver, then it is necessary to use *High (Isolated)*;

¹ The procedure is provided for IIS 5.0, but it is very similar for other versions..

7. Open the Web application configuration dialog (*Configuration...* button) and add define that Waf1 module `Wa.dll` is used for files with extensions `.waf1`, `.wfl` and `.wsvc` (or `WaDb2.dll`, if native DB2 driver is used):
 - Click the *Add* button
 - In *Executable* field enter `[Waf1Root]\Bin\Wa.dll` or `[Waf1Root]\Bin\WaDb2.dll` (or find the appropriate module using *Browse...*)²
 - In *Extension* field enter `.waf1`
 - Select *Limit* option and enter: `GET,POST`³
 - Leave *Check that file exists* empty
 - Repeat these steps for extensions `.wfl` and `.wsvc`

2.3.2. Configuring Services

The configuration of each particular service can be specified either in single common configuration file or in a separate service configuration file. No service may be configured in both common configuration file and a separate service configuration file. It is highly recommended to use separate service configuration files. The other way is supported mainly because of compatibility with older versions.

Configuration files may contain some confidential information, like username and password for database access. Thus, it is recommended to carefully define access rights on configuration files.

Configuring Services Using Separate Service Configuration Files

Waf1 service configuration files use XML syntax. The configuration file name is `www.wsvc`. It has to be located in the base service directory.

The elements of service configuration files are described in section 2.5 *Configuring Waf1 Web Service*, at page 20. An example of service configuration file is available in `Doc` subfolder of Waf1 installation.

Configuring Services Using Common Configuration File

The single common configuration file is `WAF1.Services.Ini`. It is located in base Windows directory (`C:\WINDOWS` or appropriate). An example of common configuration file `WAF1.Services.Ini.old` is available in `Windows` subfolder of Waf1 installation.

The common configuration file is used as follows:

1. The total count of configured services is specified as a value of `NOFServices` parameter, in section `[System]`. In the same section the internal names of the configured services are specified, using form: `ServiceN=Name`, where *N* is an ordinal

² Because of an IIS bug, the path to the module must not contain empty spaces.

³ If IIS version 4 is used, it is not specified what types of requests are to be handled, but the opposite - what types of requests are not to be handled by the module. Select `HEADER`, `PUT` and `DELETE`.

number of the service and *Name* is its internal name. Internal service name must not contain blank spaces:

```
[System]
NOfServices=2
Service1=WafITest
Service2=Amazon
```

- Each WafI service is configured in separate section `[Name]`, where *Name* is internal service name, e.g.:

```
[WafITest]
```

- Service parameters are specified in the form: `ParameterName=Value`.

The following parameters may be used in service configurations:

Parameter	Description
Name	Full service name. Mandatory.
Host	The base path to service Web location. It is used to distinguish different virtual servers. It consists of host name and the path specified in the location section header. Even if the location is used locally, it is necessary to use the host address specified here. Mandatory.
DefaultPage	The default page. Used only if the default page is not set in IIS. Mandatory.
DefaultContentType	Default type of MIME resources evaluated by WafI programs. It is used to create the appropriate HTTP header, if program result is not of MimeResource type. Optional. Default value is "text/html".
MaxThreads	The maximal number of WafI evaluator threads. It is recommended to set this parameter slightly above the processor count: <div style="text-align: center;"> 1 processor 4 threads 2 processors 6 threads 4 processors 8 threads </div> Optional. Default value is 4.
dirService	Base directory of WafI Web service. Usually <code>[SiteRoot]</code> . Mandatory.
dirLib	Directory with WafI library files. It is used to identify the location of common WafI libraries. Usually <code>[WafIRoot]\Lib</code> . Optional. Mandatory if common libraries are used without full path specification.
Vars	The name of the section with WafI service parameters. Each parameter is specified in form <code>Name=Value</code> . These service parameters are accessible from WafI programs by functions

Parameter	Description
	Service and ServiceValue. Here is an example: <div style="text-align: right;">Vars=WaflTestParameters</div> <div style="text-align: right;">...</div> <div style="text-align: right;">[WaflTestParameters]</div> <div style="text-align: right;">Param1=Value1</div>

Table 4: Parameters of Web service configuration for IIS Wafl module

Here is an example of common configuration file:

```

;=====
;   Basic configuration, all services specified
;=====
[System]
;service count
NOFServices=1
;service configuration section
Service1=WaflTest

;=====
;   Configuration of WaflTest service
;-----
[WaflTest]

;local service name
Name="Aplikacija Wafl Test"

;service HTTP path
Host=127.0.0.1/cgi-bin

;default page
DefaultPage=default.wafl

;path to base directory with service programs
dirService=D:\www\WaflTestApp\cgi-bin

;path to library base directory
dirLib=d:\wafl\lib\

;maximal evaluator count
MaxThreads=4

;-----
;   database

;database driver
dbDriver=ODBC

;database alias
dbAlias=WaflTest_DB

;database user name
dbUser=wafluser

;password
dbPassword=wafluserpassword

```

2.3.3. Configuring Database Connections

Database connection configuration is a part of service configuration. Any service can connect to a single database. WafI module pools database connections.

If service is configured in common configuration file, database connection is configured using parameters: `dbDriver`, `dbAlias`, `dbUser` and `dbPassword`. They correspond to parameters described in section 2.6 *Database Connections*, at page 23.

2.4. Standalone WafI Web Server

Standalone WafI Web server `WafIHttpServerWin32.exe` is simple Web server designed exclusively for development and testing purposes. It is not designed to be used in production environments.

2.4.1. Running and Configuring

Standalone WafI Web Server supports simple file transfer and processing of WafI programs. Single instance of the server can run only one WafI service. Service is configured using the same WafI service configuration files as with Apache and IIS. The configuration file has to be located in the base service directory.

The elements of service configuration files are described in section 2.5 *Configuring WafI Web Service*, at page 20. An example of service configuration file is available in `Doc` subfolder of WafI installation.

The complete procedure follows:

1. Run `WafIHttpServerWin32.exe`;
2. Specify the filename of service configuration file in Service field. The default value is correct only if there is a configuration file `www.wsvc` in the working directory of `WafIHttpServerWin32.exe`;
3. Specify the TCP port number in Port field. The default value is 4619 and should be changed only if that port is reserved for other purposes;
4. Click Start button;
5. Click Home Page button to open service home page in Web browser.

Troubleshooting

If procedure gives no expected results, try to locate the problem:

1. Switch to `WafIHttpServerWin32.exe` and click Details button to open a panel with detailed description of each request and response;
2. Switch to Web browser and try to open the site again;
3. Switch to `WafIHttpServerWin32.exe` and check if any request is registered. If no requests are reported, then either a wrong HTTP address is specified in browser, or there is a firewall preventing the services at specified port. If there are requests, the problem is probably in WafI service configuration or specific WafI programs.

After changing Web service configuration, click Reset button to stop the service, reload the configuration and restart the service.

2.4.2. Command Line Options

Standalone WafI Web Server may be used from command line using following syntax:

```
wafIHttpServer <options>
```

where <options> is blank separated options list, which may include the following options:

- <cfgFile> – configuration file name, begins with a letter;
- <portNumber> – port number, integer. If not specified the default is 4619;
- start – automatically start the service;
- open – automatically start the service and open home page;
- minimize – minimize server window.

2.5. Configuring WafI Web Service

WafI service configuration files use XML syntax. Recommended configuration file name is `www.wsvc`, and recommended configuration file location is the base service directory. If MS IIS is used, both recommendations are mandatory.

One configuration file configures one service.

Configuration file may contain some confidential information, like username and password for database access. Thus, it is recommended to carefully define access rights on configuration files.

An example of service configuration file is available in `Doc` subfolder of WafI installation.

2.5.1. Configuration File Options

The following XML tags are available for use in configuration files:

Tag/Attribute	Description
<WafIService	Base service description tag that identifies the service. Mandatory. Outermost tag.
name="..."	Service name. Mandatory.
type="WWW"	Service type. Optional. Default value is "WWW".
description="..."	Short service description. Optional. Default value is empty string.
<WWW>	Configuration of Web options. Bounds all Web specific configuration parameters.

Tag / Attribute	Description
	Mandatory for services of "WWW" type.
<code><Host name="..." /></code>	Host name for the service. Optional. Can be specified many times, if many names or IP addresses are dedicated to the server.
<code><Basedir path="/home" /></code>	Base URI path to the service, not including the host name. It is used if the service is defined not as a complete Web site, but as a virtual directory of a larger site. Optional. Default value is "/".
<code><Directories</code>	Locations of specific directories. Mandatory.
<code>serviceRoot="path"</code>	Base service directory. Optional. Default value is the directory in which the configuration file is located.
<code>systemLib="path"</code>	WafI system libraries base directory. Usually <code>[WAFI_PATH] / Lib</code> . Mandatory.
<code>serviceLib="path"</code>	WafI service libraries base directory. Mandatory.
<code><Default</code>	Configuration of default parameters. Mandatory.
<code>file="default.wafI"</code>	Default service file. Optional. Not used with IIS.
<code>mimeType="text/html"</code>	Default type of MIME resources evaluated by WafI programs. It is used to create the appropriate HTTP header, if program result is not of MimeResource type. Optional. Default value is "text/html".
<code><Database</code> <code>driver="DB2"</code> <code>alias="WafIHome"</code> <code>username="wafIuser"</code> <code>password="wafIpswd"</code>	Database connection configuration. It is described in details in section 2.6 <i>Database Connections</i> , at page 23. Optional. All attributes are mandatory.
<code><Evaluators</code>	Configuration of WafI evaluators. Optional.

Tag/ Attribute	Description						
maxthreads="4"	The maximal number of Wafl evaluator threads. It is recommended to set this parameter slightly above the processor count: <table> <tr> <td>1 processor</td><td>4 threads</td></tr> <tr> <td>2 processors</td><td>6 threads</td></tr> <tr> <td>4 processors</td><td>8 threads</td></tr> </table> Optional. Default value is 4.	1 processor	4 threads	2 processors	6 threads	4 processors	8 threads
1 processor	4 threads						
2 processors	6 threads						
4 processors	8 threads						
<Debug	Debugging configuration. Optional.						
on="1"	Set debugging mode on (1) or off (0). In debugging mode error reports are more detailed. Optional. Default value is 0.						
errLogFileBase="..."	The full path and the beginning of the name of log files. The name extends with date and thread information. Optional.						
<Parameters	Configuration of environment and service parameters. Optional.						
<Param name="..." value="..." />	Definition of service parameter name and value. Optional. Can be specified many times.						
<EnvParam name="..." value="..." />	Definition of environment name and value. Optional. Can be specified many times.						

Table 5: Tags and attributes in Wafl service configuration file

An Example of Wafl Service Configuration File

```

<WaflService
  name="WaflHomePage"
  type="WWW"
  description="Wafl project home page."
>

<WWW>
  <Host name="www.waflproject.org" />
  <Host name="127.0.0.1" />
  <Host name="192.168.0.3" />

  <Basedir path="/home" />

  <Default
    file="default.wafl"
    mimetype="text/html"
  />

</WWW>

<Directories
  serviceRoot="D:\WaflSite\wwwroot"

```

```

        serviceLib="d:\wafl\lib"
        systemLib="d:\wafl\lib"
    />

    <Database
        driver="DB2"
        alias="WafHome"
        username="wafluser"
        password="wafluserpassword"
    />

    <Evaluators
        maxthreads="4"
    />

    <Debug
        on="1"
        errLogFileBase="d:\wafl\log\[WafSite]"
    />

    <Parameters>
        <Param name="testvar1" value="testvalue1" />
        <Param name="testvar2" value="testvalue2" />
        <!-- DB2 code page. -->
        <EnvParam name="DB2CODEPAGE" value="1250" />
    </Parameters>

</WafService>

```

2.6. Database Connections

Database connections are realized by database drivers embedded in Waf modules:

- Database connections on Linux are realized by ODBC driver, supported by unixODBC library. The unixODBC library is required.
- Database connections on Windows are realized by generic ODBC driver and native DB2 drivers.

Database connection is configured as a part of Waf service configuration. Connections are pooled by Waf module, which opens and closes connections when it is appropriate.

2.6.1. Database Connection Configuration

Single Waf service can access single database, using single user account. The same stands for single command line programs.

Database access is configured using the following parameters:

Parameter	Description
dbDriver	Database driver. Supported drivers are: <ul style="list-style-type: none"> • ODBC – For DB connections using ODBC. Windows version uses Windows ODBC drivers. Linux version uses unixODBC library; • DB2 – Provides full native connections to IBM DB2 databases.

Parameter	Description
	<p>Each major version of DB2 should use specific software build. Supports ODBC connections to other databases, too, but some compatibility issues may arise.</p> <ul style="list-style-type: none"> None – The "void" driver. No database connection is used. If this "driver" is specified then other parameters are not significant. <p>Mandatory if a database is used. Default value is "None".</p>
dbAlias	<p>Database name. Depends on used driver (<i>dbDriver</i>):</p> <ul style="list-style-type: none"> for ODBC, <i>dbAlias</i> value is ODBC DSN of the database; for DB2, <i>dbAlias</i> value is DB2 database name. <p>Mandatory if a database is used.</p>
dbUser	<p>User account name for database connection.</p> <p>Optional. Default value is empty string.</p>
dbPassword	<p>User account password for database connection.</p> <p>Optional. Default value is empty string.</p>
dbKeepSessions	<p>The allowed count of pooled connections. If 0 is specified, then no limit is configured.</p> <p>Optional. Default value is 0.</p> <p><i>Not supported in current version.</i></p>
dbQueryIsolation	<p>The default query isolation level. Allowed values are <i>s</i> (<i>Serializable</i>), <i>rr</i> (<i>Repeatable Read</i>), <i>rc</i> (<i>Read Committed</i>) and <i>ru</i> (<i>Read Uncommitted</i>).</p> <p>Optional. Default value is <i>ru</i>.</p> <p><i>Not supported in current version.</i></p>
dbTransIsolation	<p>The default transaction isolation level. Allowed values are <i>s</i> (<i>Serializable</i>), <i>rr</i> (<i>Repeatable Read</i>), <i>rc</i> (<i>Read Committed</i>) and <i>ru</i> (<i>Read Uncommitted</i>).</p> <p>Optional. Default value is <i>rr</i>.</p> <p><i>Not supported in current version.</i></p>

Table 6: Database connection configuration parameters

2.6.2. Specific Issues

IBM DB2 and Apache for Linux

The process that establishes the connection (the Waf kernel) must have information on DB2 server instance. That is achievable by setting the DB2INSTANCE environment variable, using Apache configuration directive `SetWafEnv`. The value of the variable must be the name of the DB2 instance. For example, if the instance name is `db2inst1`, than insert the following directive in the Waf service configuration section:

```
SetWafEnv DB2INSTANCE db2inst1
```

The current implementation of WafI module is constrained by global visibility of this parameter. Only the last set value is considered. Thus, all services share its value and can use only one DB2 instance.

IBM DB2 in Environments With Many Code Pages

If DB2 database is used, where database code page is not the same as the default system code page, the connection may fail. In that case, it is necessary to set the DB2 environment variable DB2CODEPAGE to database code page. The configuration of the variable depends on the used WafI software:

- Command line WafI c1WafIDb2 has command line option `-env`:

```
-env:DB2CODEPAGE=1250
```

- If Apache configuration file is used, then `SetWafIEnv` directive sets appropriate value:

```
SetWafIEnv DB2CODEPAGE 1250
```

The current implementation of WafI module is constrained by global visibility of this parameter. Only the last set value is considered. Thus, all services share its value and can use only one DB2 instance.

- If WafI service configuration file is used (for either Apache or IIS), the value of DB2CODEPAGE variable is configured by insertion of following tags:

```
<Parameters>
  <EnvParam name="DB2CODEPAGE" value="1250" />
</Parameters>
```

The current implementation of WafI module for Apache is constrained by global visibility of this parameter. Only the last set value is considered. Thus, all services share its value and can use only one DB2 instance.

In the case of IIS, it is necessary to set high service isolation (section 2.3 *WafI Module for MS IIS*, at page 15) to limit the scope of the parameter to single service.

ODBC and DB2 for Linux

To use DB2 database using ODBC driver it is necessary to follow the procedure:

1. Install unixODBC;
2. Configure unixODBC either by running `ODBCConfig` (from command line) or by editing files `/etc/odbcinst.ini` and `/etc/odbc.ini`:
 - add DB2 driver (e.g. version 8.1):

```
Name = DB2
Description = DB2 Driver
Driver = /opt/IBM/db2/V8.1/lib/libdb2.so
Setup = /opt/IBM/db2/V8.1/lib/libdb2cfg.so
FileUsage = 1
DontDLClose = 1
```

- define ODBC DSN for particular database:

```
Name = dbname
Driver = DB2
```

ODBC and BLOB Data

BLOB data types are used with ODBC driver in a same way if they are textual types. The single limit is the process address space. The driver reserves the exact memory for the data it reads (not the maximal allowed for the column). However, some ODBC drivers introduce specific limits on BLOB reading. In that case, try to avoid the problems by specifying BLOB columns at the end of the column list in SELECT clause.

BDE and BLOB Data

Legacy services uses BDE DB driver.

BLOB data types are used with BDE driver as textual data. However, there is a limit in both reading and writing of BLOB data. Program must read BLOB data in parts of up to 32000 bytes, and join the parts after the reading. If program writes BLOB data larger than 32000 bytes, the names of corresponding transactional functions arguments must begin with "blob_".

2.7. Logs

Web server versions of WafL support logging of problems and significant actions.

2.7.1. Log Files Location for Windows

If WAFL_PATH environment variable is set and its value is the path to a folder with Log subfolder, then log files are created in folder [WAFL_PATH] \Log. Otherwise, log files are created in folder C:\.

2.7.2. Log Files Location for Linux

If WAFL_PATH environment variable is set and its value is the path to a folder with Log subfolder, then log files are created in folder [WAFL_PATH] /Log. Otherwise, log files are created in folder /tmp/⁴.

2.7.3. Log Files Names

After each Web server restart a new log file is created. The log file name has the form:

```
waf1-<year>.<month>.<day>-<count>.log
```

For example:

```
waf1-2004.04.23-9.log
waf1-2004.04.23-10.log
```

⁴ If Apache Web server is used, the environment variable WAFL_PATH may be specified in Apache configuration file (section *Configuring Services Using Apache Configuration File*, at page 13).

2.8. Development Tools

The Waf1 distribution includes some resources allowing the customization of operating system and text editors.

2.8.1. Optional Configuring of Windows

To register Waf1 file types with operating system, follow the following steps:

1. There is `waf1.reg` file in `Support` folder of Waf1 installation. This file contains registrations for Waf1 file types. Open this file in a text editor;
2. Change item:

```
[HKEY_CLASSES_ROOT\Waf1Program\shell\Run\command]
@="\"c:\\waf1\\bin\\clwaf1.exe\" -wait \"%1\""
```

to reflect the actual path to `clWaf1` program;

3. In all items of the form:

```
[HKEY_CLASSES_ROOT\...\shell\Edit\command]
@="notepad.exe \"%1\""
```

it is possible to replace `notepad.exe` program with another text editor;

4. In all items of the form:

```
[HKEY_CLASSES_ROOT\...\DefaultIcon]
@="c:\\winnt\\system32\\SHELL32.dll,..."
```

it is possible to specify another icon files;

5. Save the file and merge it to Windows registry (right click/Merge).

2.8.2. Text Editor Customization

The `Support` folder of Waf1 installation contains files for customization of some popular text editors:

nalaze se datoteke za obezbeđivanje raspoznavanja sintakse programskog jezika Waf1 u nekim programima za uređivanje teksta:

- File `wordfile.txt` contains the description of Waf1 syntax for UltraEdit text editor. Find `<UltraEdit_install_dir>\wordfile.txt`, open it in text editor and append the complete content of `wordfile.txt` at the very end of the file. If necessary, change the ordinal number at the beginning of Waf1 syntax description: `/L6"Waf1"`.
- File `waf1.syn` contains the description of Waf1 syntax for TextPad⁵ text editor. Copy the file to `<TextPad_install_dir>\Samples`. Open *TextPad* and select `Configure\New Document Class` command from menu. Set the new class name to "waf1" and add extensions (class members) `*.waf1, *.wfl, *.wlib`.

⁵ Thanks to Miloš Babić.

Index

A

- AddHandler 9
- answerAction 6
- Apache
 - configuration 9,12
 - Linux 7
 - services 10
 - setup 7,9
 - Windows 9
- ask 6

B

- BDE 22

C

- cmdLineArgs 6
- command line interpreter
 - limitations 6
 - Linux 4
 - setup 4
 - using 5
 - Windows 4
- command line options 5
- common configuration file 15
- ConfigFile 11
- configuration
 - Apache 12

D

- database
 - BDE 24
 - BLOB 24
 - code pages 23
 - connections 21
 - DB2 23
 - drivers 3
 - IIS 13,17
 - Linux 24
 - ODBC 24

- Database 19
- DB2 22
 - code pages 23
 - Linux 23
- DB2CODEPAGE 23
- DB2INSTANCE 23
- dbAlias 22
- dbDriver 22
- dbKeepSessions 22
- dbPassword 22
- dbQueryIsolation 22
- dbTransIsolation 23
- dbUser 22
- Debug 19
- DefaultContentType 12
- DefaultMimeType 16
- DefaultPage 16
- Directories 18
- dirLib 16
- dirService 16

E

- Evaluators 18

F

- Form 6
- FormValue 6

H

- Host 11,16
- httpd.conf 7,9,11
- httpHost 6
- httpPath 6
- httpScript 6

I

- IIS
 - services 15
 - setup 13
- input 6

L

LibDirectory 12
libraries 4
Linux
 Apache 7
 command line interpreter 4
 setup 7
LoadModule 8, 10
Location 9, 11
log files 24

M

MaxRequestsPerChild 7, 8, 10
MaxThreads 12, 16
MaxThreadsPerChild 8
mod_wafl 8, 9, 10

N

Name 15

O

ODBC 22

P

Parameters 19
perchild.c 8
prefork.c 7, 10

R

RESET 6

S

service
 configuration 10
Service 6
service configuration file 11
services
 configure 15, 18
ServiceValue 6

Session 6

SessionValue 6

SET 6

setup 3

 Apache 7, 9

 IIS 13

 Linux 7

 Windows 9

SetWafEnv 12

SetWafParam 12

standalone Web server 17

T

TextPad 26

ThreadsPerChild 8, 10

tools 25

U

UltraEdit 26

V

Vars 16

versions 3

virtual directory 10

W

WafL.Services.Ini 15

WAFL_PATH 4, 5

WafService 11, 18

Web location 10

Windows

 Apache 9

 command line interpreter 4

 IIS 13

 setup 9

worker.c 8

WWW 19

www.wsvc 11, 18