FaMe
Facilities Management
Software GmbH

Installation guide

FaMe®-FM V5

FaMe IIS Plugin Version 3.0

(FameHandler.dll)

For IIS 8 and 10

on Windows Server 2012, 2016, and 2019
covers Oracle 12 and 19

Includes Setup of FaMe PDF Writer
and FaMe graphics module
on IIS

Revision 1.10

9.12.2021

Revisions

0.1	M.Eisenhardt	11.8.2017
0.2	M.Eisenhardt	4.12.2017
	Title change, TNS configuration to English, Plugin config section completed	
1.0	M.Eisenhardt	12.12.2017
1.1	M.Eisenhardt	19.06.2018
	Revised, added section about PDF writer	
1.2	identical to Rev. 1.1	
1.3	M. Eisenhardt	14.8.2018
	added section on barcode modules for Perl, script *start_pdf_report.pl*, script *pdf_report.check_modules.pl*, the log directory, PDF writer testing, and graphics module. Parameter *system_charset* in *pdf_report.conf* file now has standard value *UTF-8*.	
1.4	M. Eisenhardt	15.8.2018
	Added section about PDF Writer troubleshooting	
1.5	M. Eisenhardt	11.4.2019
	Revised introduction and web.config section Added section about the use of oraprovcfg.exe to install DLLs in the GAC Added quick installation checklist	
1.6	M. Eisenhardt	15.4.2019
	Added quick installation guides for PDF writer and graphics module	
1.7	M. Eisenhardt	9.7.2019
	changed description of parameter requestValidationMode	
1.8	M. Eisenhardt	25.7.2019
	fixed table layout in chapter 2.1/Overview	
1.9	M. Eisenhardt	29.11.2021
	Added new sections: 2.6 Tuning the FaMe IIS plugin for performance 2.7 Debug log	
1.10	M. Eisenhardt	9.12.2021
	Added new section: 2.8 Setting up the FaMe IIS Plugin for Windows Single-Sign-on (SSO) configuration	

Contents

FaMe-FM Release 5:
Installation and configuration of the FaMe-IIS-Plugin (FameHandler.dll) for IIS 8 on Windows Server 2012/2016/2019
plus
Installation and configuration of the FaMe PDF Writer and Graphics Module for IIS on Windows

Audience

This document is intended for use by system administrators that are going to install the FaMe IIS Plugin and/or the FaMe PDF Writer or Graphics Module on Windows 2012 systems.

About this document

This document covers the installation of FaMe IIS Plugin version 3.0 for Oracle 12c on Windows Server 2012:

- Basic setup of the IIS
- Oracle client installation and configuration
- Setup and configuration of the FaMe IIS plugin
- Setup of the FaMe PDF writer (external program running as Perl CGI script)
- Basic function tests

Please note: This document basically applies as well to Oracle 19c installations on Windows Server 2016 and 2019. The FaMe IIS Plugin is available for Oracle 19c as well.

How to use this document

This document explains the installation of the software components in detail and at length. Parts of it may not be needed for the planned installation and are intended for use as a reference.

The chapter *1 The FaMe IIS Plugin (FameHandler.dll)* is a short introduction to the Fame IIS plugin.

The chapter *2.1 Overview* provides information about the chapters to refer to, depending on the functions to install.

For the **quick installation** of the FaMe IIS Plugin, the FaMe PDF Writer, and the FaMe Graphics Module there are checklists in chapter *2.2 Quick installation guides*.

1 The FaMe IIS Plugin (FameHandler.dll)

The FaMe IIS Plugin provides the interface between webserver and database for the web-based FaMe FM 5 database application.

FaMe-FM Version 5 is a web-based database application for Oracle databases. The application functions are written in the PL/SQL language and are installed and run inside the database. The FaMe IIS Plugin translates CGI requests for its configured virtual path into PL/SQL calls for the database.

A call to

```
http://server/path2app/fame/mypackage.the_procedure?param1=value1&param2=value2
```

is translated by the FaMe IIS Plugin to a procedure call:

```
begin
  mypackage.the_procedure(param1=>'value1', param2=>'value2');
end;
```

and sent to the database for execution. The result produced by the procedure is retrieved from the database by the plugin and returned to the web client; in most cases a browser.

Please note that in any case the virtual path to the database has exactly 2 directory levels of which the 2nd is always *fame*:

```
http://server/path2app/fame/
```

The 1st directory in the path, `path2app`, may be renamed to any valid URL directory name you choose. The 2nd level `fame` is required by the FaMe application internally and must not be changed.

It is possible to configure several database URLs on the server, for instance, test and production:

```
http://server/fame_prod/fame/
```

and

```
http://server/fame_test/fame/
```

may be on the same server.

The configuration is done via the *web.config* file. See chapter *2.5.8 web.config file*.

In the database the *Oracle PL/SQL Web Toolkit* is used as application framework. It is part of the Oracle database's standard set of packages.

This plugin also provides the ability to use integrated Windows authentication ("single sign-on") for the FaMe application, allowing for Windows users to log in to FaMe with their Windows accounts and current Windows session. To actually use integrated Windows authentication with the FaMe database application further settings have to be applied inside the FaMe application.

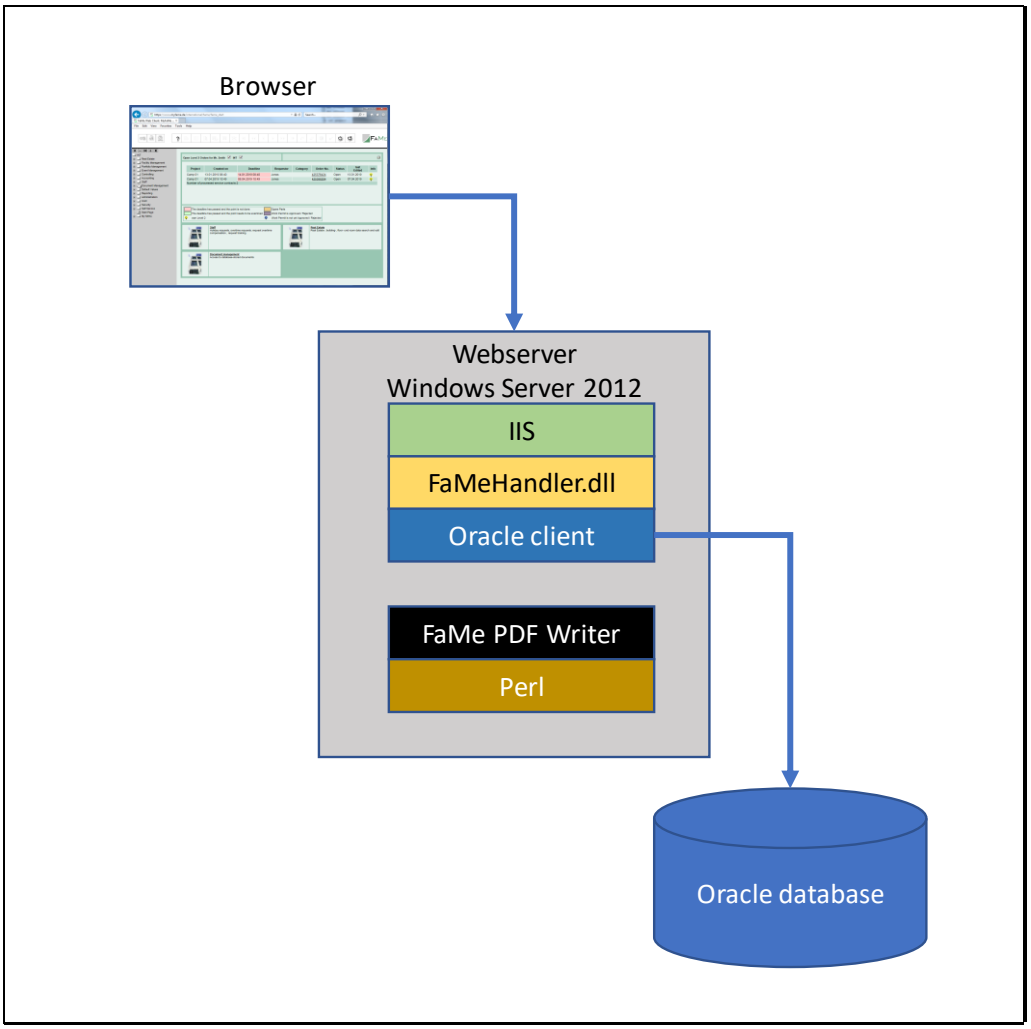
The FaMe IIS Plugin is set up as IIS handler for virtual paths mapped to databases. The resources inside these paths are the PL/SQL procedures in the database schemas the plugin connects to.

The plugin is provided as a DLL. It requires the .net 4.0 library and a 64 bit Oracle 12c client installation on the webserver.

1.1 Setup described in this document

This document illustrates how to

- Install and set up the Microsoft Internet Information Server (IIS) on Windows Server 2012 for use with the FaMe FM V5 database application
- Install and configure the FaMe IIS Plugin (FameHandler.dll) and the Oracle client needed to run this plugin
- Install and configure the FaMe PDF Writer and the required Perl interpreter
- Install and configure the FaMe Graphics Module and the required Perl interpreter



FaMe IIS Plugin and PDF Writer setup

After a successful setup the FaMe VM V5 application should be available via the webserver. In most places throughout this document the URL for the application is assumed to be

```
http://server/path2app
```

with the virtual path /path2app being used as an example.
With this example URL the virtual path to FaMe database procedures is

```
http://server/path2app/fame/<package.procedure>
```

e.g.
`http://server/path2app/fame/std_util_pk.login`

The FaMe PDF Writer will have this URL using the example virtual path:

```
http://server/path2app/perl/pdf_report.pl
```

The FaMe Graphics Module will have this URL using the example virtual path:

```
http://server/path2app/perl/fm5_chart.pl
```

Files for the application that reside on the webserver will be in these virtual paths:

```
http://server/path2app/html/
http://server/path2app/image/
http://server/path2app/js/
http://server/path2app/jquery/
```

and possibly some more.

1.2 System requirements

Please note: this section does neither cover Windows Server 2016 and 2019 nor Oracle 19c currently.

Concerning the Perl interpreter installations using Strawberry Perl are supported but not yet documented.

The installation of the FaMe IIS plugin requires

- FaMe IIS plugin version 3.0.12102.X or 3.0.12201.X, 64 Bit, depending on the Oracle software version installed
- Windows Server 2012 Standard R2 (Windows Server 2012 should be working but have not yet been evaluated)
- IIS 8
- Microsoft .net framework 4.0 or higher, 64 bit
- Oracle-Client 12.1.0.2 or 12.2.0.1, 64 Bit with these components:
    - Oracle Net
    - Oracle Data Provider for .NET
    - Oracle Data Provider for ASP.NET

  If Oracle Server software is installed on the machine in question a separate client is not needed. If the Oracle software installed is of any other version, contact FaMe support to get the FaMe IIS Plugin's version compiled for that Oracle software version.

  The FaMe PDF writer -if it is to be installed- requires

- FaMe PDF writer Perl library
- Perl interpreter: use Active State Perl here. If PDFlib 9.1 and ChartDirector 5.1 are in use together (i.e. PDF Writer *and* Graphics Module*)*, ActivePerl 5.20 is the only Perl interpreter version supporting both. If no Graphics Module is to be installed use ActivePerl 5.24.
- *PDFlib* library (PDF Writer only, commercial, to be licensed individually per server) current version is 9.1.2
- *ASE Perl ChartDirector* library if graphical reporting functions are in use (graphics

module only, commercial library, contact FaMe support for license)
version 5.1

1.3 Version history of the FaMe IIS Plugin

The version history is provided as a separate text file named *Versions-FaMe-IIS-Plugin.txt*.

2 Installation

2.1 Overview

For installation of the FaMe IIS plugin the following parts of this document apply:

Chapter (click onto the links below to get there)	When to apply
***2.3** Adding the .net 4.5 feature and webserver role (IIS) to Windows Server 2012* (Page 13)	Always
2.4 Oracle client installation (Page 20)	Only if the installation is *not* done on a server with Oracle Server installed, but you may refer to section 2.4.6 TNS configuration for database connections vs. EZCONNECT for configuration and testing of the Oracle database connection in either case.
2.5 IIS configuration for the FaMe IIS Plugin and setup for the first database connection (Page 33)	Always
2.5.8.4 Second FameHandler definition for the FaMe PDF Writer (Page 41)	Only if the FaMe PDF Writer is installed as well *and* Integrated Windows authentication is used
2.10 Setting up the FaMe Graphics module (Page 79)	Only if the FaMe Graphics Module is to be installed

2.2 Quick installation guides

2.2.1 Quick installation of the FaMe IIS Plugin

To install the FaMe IIS Plugin quickly

- read chapter 1 to get a basic idea of the FaMe IIS plugin
- configure the IIS according to chapter *2.3 Adding the .net 4.5 feature and webserver role (IIS) to Windows Server 2012* (page 13).
- If neither an Oracle client nor Oracle server is installed, install an Oracle client version 12.1 or 12.2. If you are familiar with Oracle clients, you may skip chapter *2.4 Oracle client installation*.
- Make sure the Oracle ODP.NET (*oracle.dataaccess.dll*) and ASP.NET libraries (*oracle.web.dll*) are installed properly. Refer to chapter *2.4.5 ODP.net libraries and their registration in the Windows Global Assembly Cache (GAC)*
  Alternately, copy these DLLs into the webserver's *bin* directory.
- Choose a virtual path for your FaMe FM application and create a directory for it under the IIS's document root (usually *c:\inetpub\wwwroot*). Unzip the package of static application files for your FaMe FM application into that directory.
  Example: *c:\inetpub\wwwroot\path2app*

Throughout this document this application directory has been named ***path2app*** as an example. You will most probably choose another name.

- Follow chapter *2.5 IIS configuration for the FaMe IIS Plugin and setup for the first database connection* to get the FameHandler.dll installed and configured.
- Try to call both the *getVersion* page and a database procedure as described in chapter *2.5.9 Basic test of the FaMe IIS Plugin*
- If this doesn't work refer to chapter *2.5.10 Troubleshooting: If the FaMe IIS Plugin doesn't work*

2.2.2 Quick installation of the FaMe PDF Writer

First decide whether the FaMe Graphics Module is to be installed as well.
If yes: install Active Perl 5.20
If no: install any one of Active Perl 5.20, 5.22, or 5.24

You need

- *ActivePerl* 5.20, 5.22, or Perl 5.24 – see previous paragraph. Use a 64-bit version.
- *PDFlib* version 9 for Perl, 64 bit
- *ASE ChartDirector* library for Perl if you want to install the graphics module as well
- FaMe PDF Writer library package
- FaMe PDF report template package for your application
- FaMe IIS plugin installed and configured for your database

Installation procedure for the PDF Writer:

- The FaMe PDF Writer requires the FaMe IIS Plugin as gateway to the database. See previous section for quick installation procedure.
- Install ActivePerl 5.20 or 5.24; select version as explained above
- Extract the FaMe PDF Writer installation package (zip format) into a temporary directory of your choice. Copy the folder *FaMe_FM5* to the installation package (it's inside the *lib* folder) to *c:\Perl64\site\lib*. There should be a folder *c:\Perl64\site\lib\FaMe_FM5* now.
- Copy 2 files from the PDFlib installation package into the dame *site\lib* folder:
  bind\perl\pdflib_pl.pm from PDFlib into *c:\Perl64\site\lib*
  If you have ActivePerl 5.20:
  bind\perl\ ASperl520 \pdflib_pl.dll from PDFlib into *c:\Perl64\site\lib*
  If you have ActivePerl 5.24:
  bind\perl\ ASperl524 \pdflib_pl.dll from PDFlib into *c:\Perl64\site\lib*
- Copy the files *pdf_report.pl* and *pdf_report.conf* from the *cgi* directory in the installation package to the *perl* subfolder of the FaMe application directory.
- Extract the package with PDF report templates for your FaMe application into the subdirectory *html/tmpl* of the FaMe application directory.
- Edit the *pdf_report.conf* you copied into the *perl* subfolder. A minimal *pdf_report.conf* file is shown in section *2.9.11 Basic PDF Writer configuration*.
- Create a directory *logs* under the *perl* folder and set full control permission for the IUSR user as this directory is used for logs and temporary files.
- In the IIS manager convert the perl folder to an application, set anonymous access,

and configure a script mapping for .pl files using the Perl interpreter. See *2.9.13 IIS configuration for Perl scripts* (page 70) for instructions.
Instead of setting the script map in IIS manager you may copy the web.config file shown in that chapter into the *perl* folder but check the paths inside the file for correctness.
Converting the *perl* folder to an application must done in any case.

- You might add the optional Perl module *GD::Barcode* and *Barcode::Code128*. This is covered in section 2.9.9Barcode libraries GD::Barcode and Barcode::Code128 (Page 67)
- To test the FaMe PDF Writer, open the FaMe application in the browser and log in as an administrative FaMe user, preferably as the *Egon* user, provided you have the password. Follow the instructions in chapter *2.9.14 Testing the FaMe PDF Writer* (page 75).

2.2.3 Quick installation of the FaMe Graphics module

You need

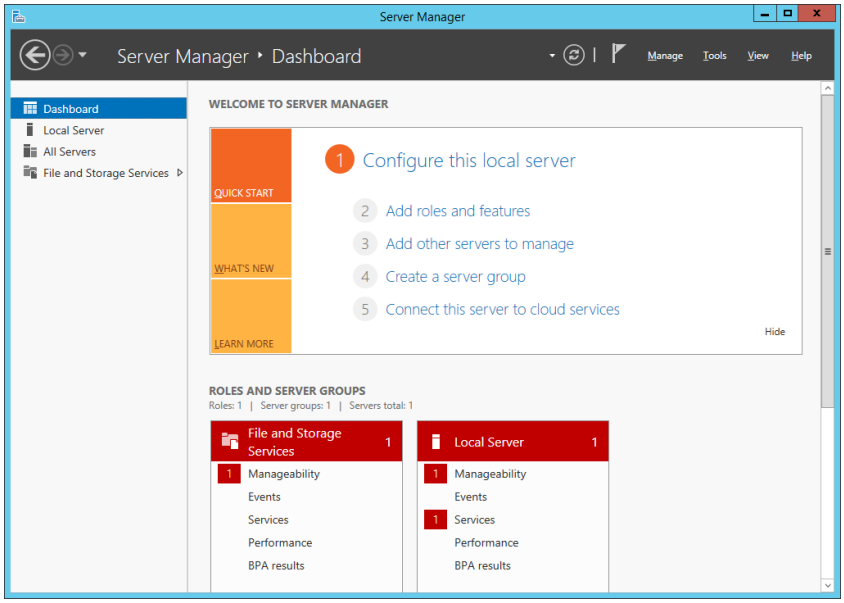
- *ActivePerl* 5.20, 64-bit
- *ASE ChartDirector* library 5.1 for Perl
- FaMe PDF Writer library package
- Oracle client installed and configured

Installation procedure for the FaMe Graphics Module

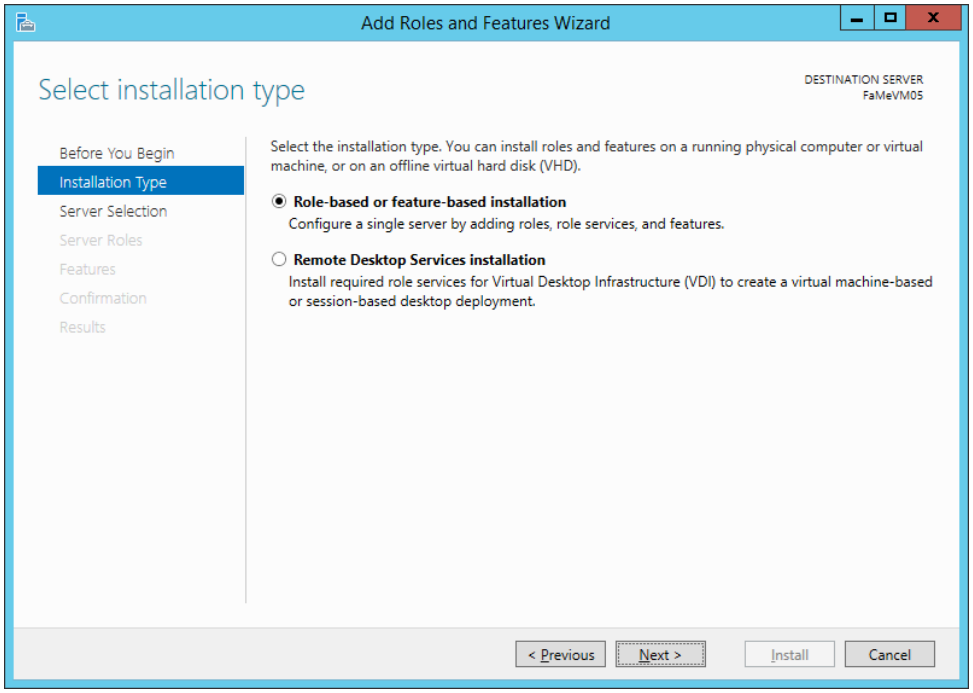
- **If you already installed the FaMe PDF Writer skip these 3 steps**, otherwise, they to be done now:
  - Install *ActivePerl* 5.20
  - Extract the FaMe PDF Writer installation package (zip format) into a temporary directory of your choice. Copy the folder *FaMe_FM5* to the installation package (it's inside the *lib* folder) to *c:\Perl64\site\lib*. There should be a folder *c:\Perl64\site\lib\FaMe_FM5* now.
  - Create a directory *logs* under the *perl* folder, set anonymous access, and set full control permission for the IUSR user as this directory is used for logs and temporary files.
- Open the *Perl ChartDirector* installation package and copy all files from the zip's *ChartDirector\lib* folder into *c:\Perl64\site\lib*
- From the FaMe PDF Writer's installation package copy these 2 files into the application's *perl* folder:
  fm5_chart.pl
  fm5_chart.conf
- Edit the configuration file fm5_chart.conf according to section *2.10.4 Configuration file fm5_chart.conf* (page 80). You will need the database's host name or EZCONNECT data to set the database connection. Database username is always FAME_WEB.
- Test the graphics module according to section 2.11 Testing the FaMe Graphics Module (page 81)

2.3 Adding the .net 4.5 feature and webserver role (IIS) to Windows Server 2012

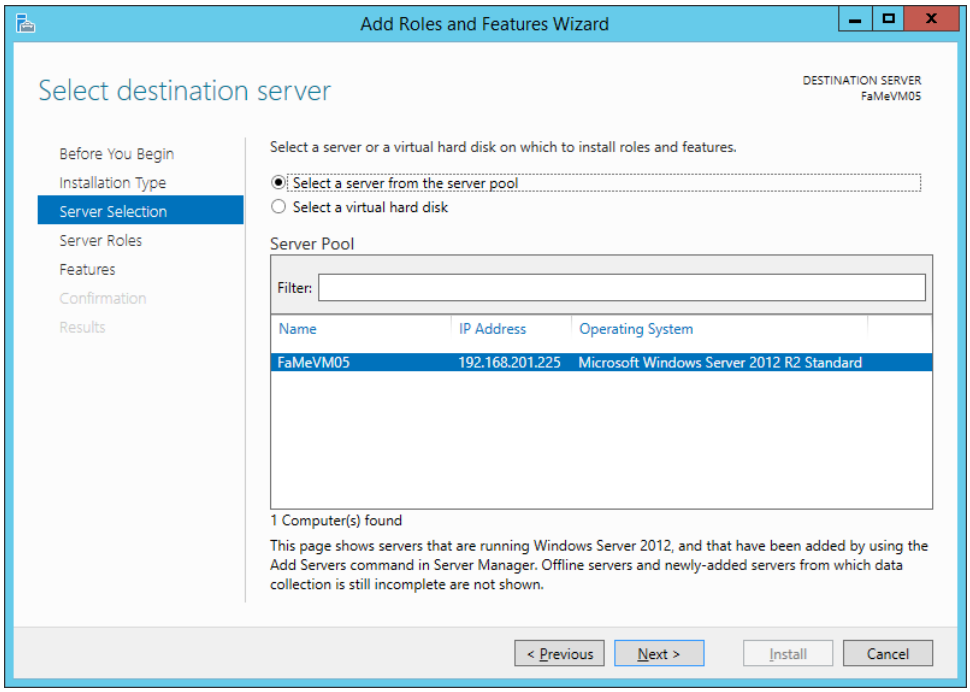
The IIS is a server role that has to be activated using the server manager.



Select *Add roles and features* and skip the next page titled „*before you begin*".
Select *role based or feature based installation* on the consecutive page:

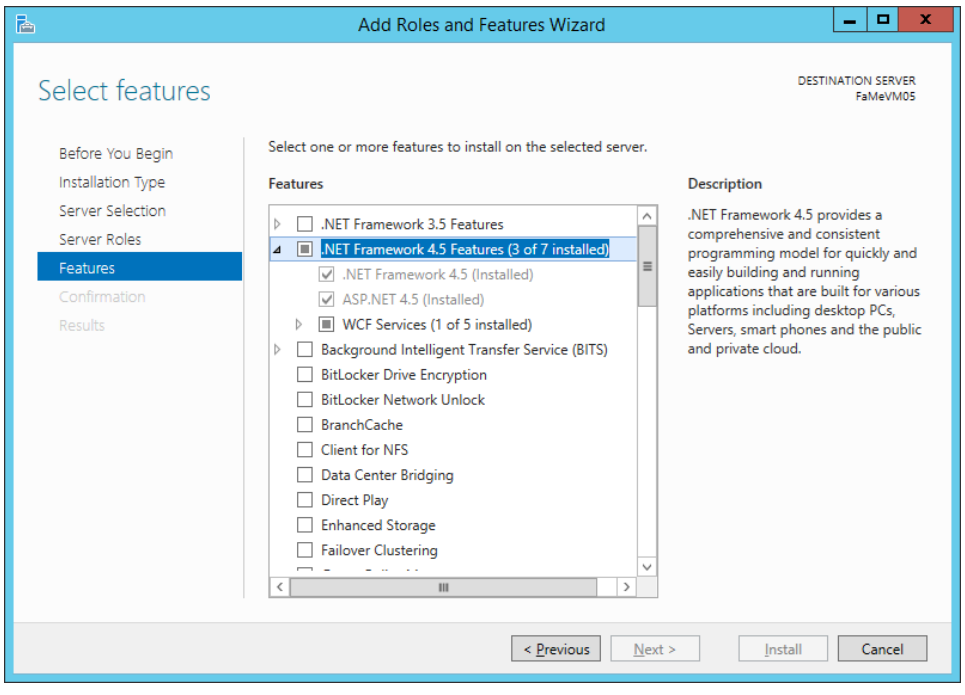


Select the server you are about to modify:



2.3.1 The .net 4.5 framework feature

The .net 4.0 framework is required for the FaMe IIS plugin. Windows Server 2012 provides version 4.5 as an installable feature:



This feature installation will be handled automatically with the installation of the ASP.net 4.5 feature as shown in the next section. The *WCF services* option is not changed here.

2.3.2 IIS role and features

Select the role to add (IIS). Continue even if IIS is already installed as there may be IIS

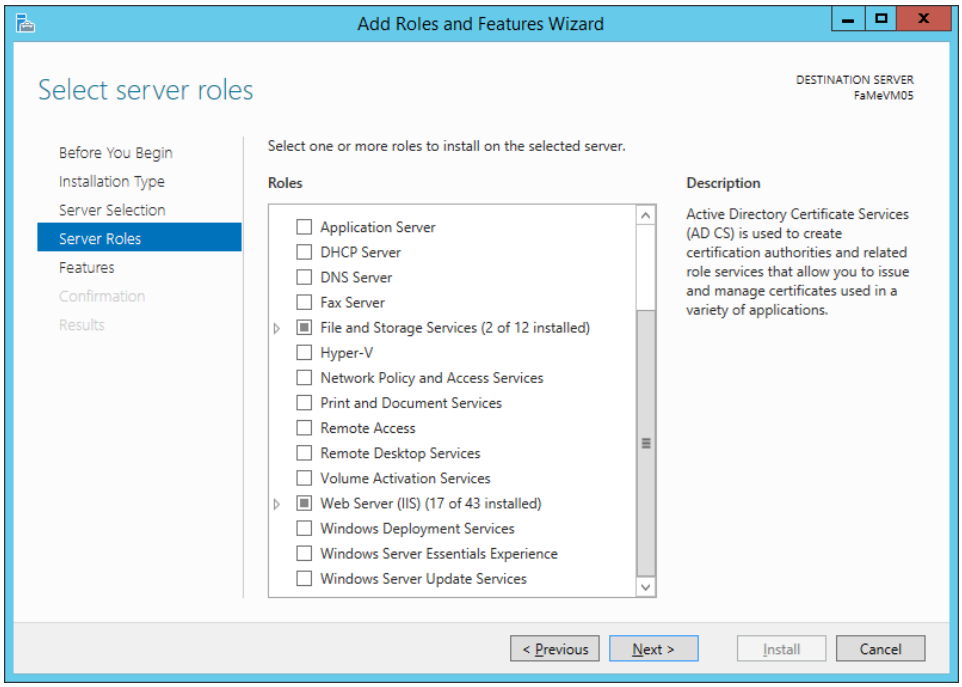
features not yet activated. Basically, these IIS features are required:

ISAPI extensions

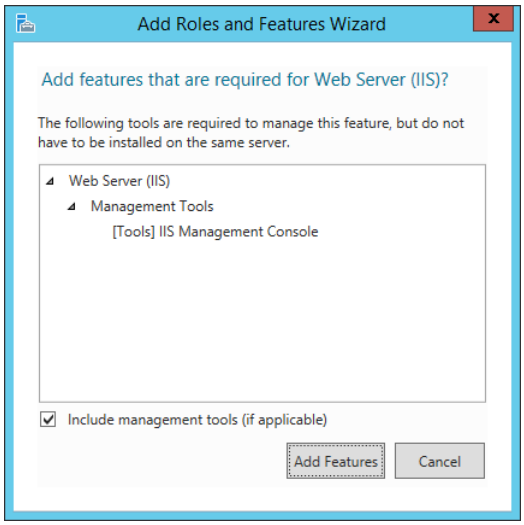
ASP.net 4.5

CGI scripting

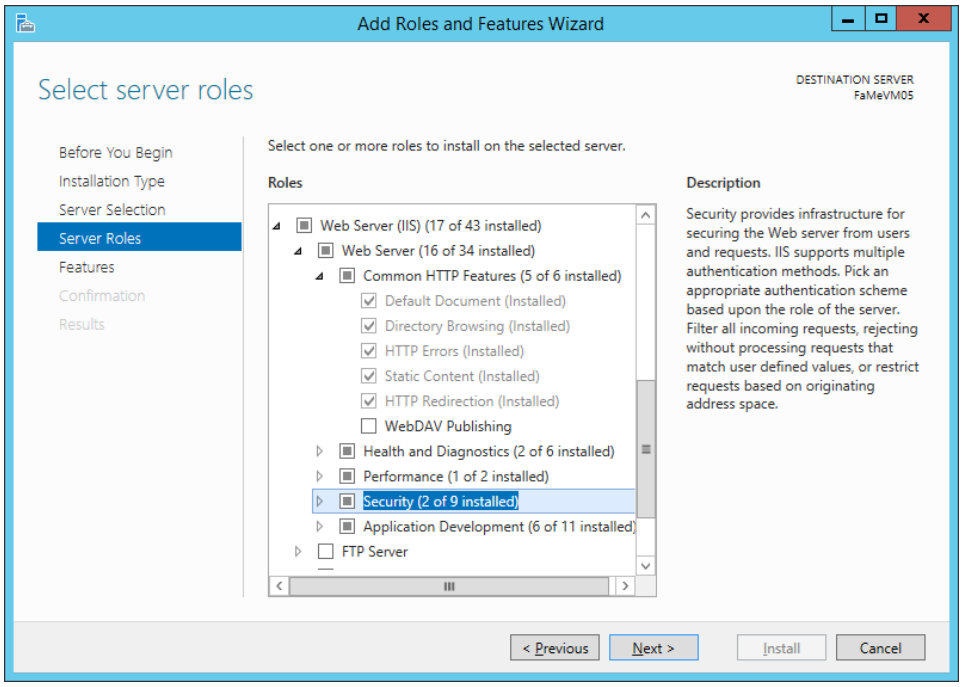
Integrated Windows authentication

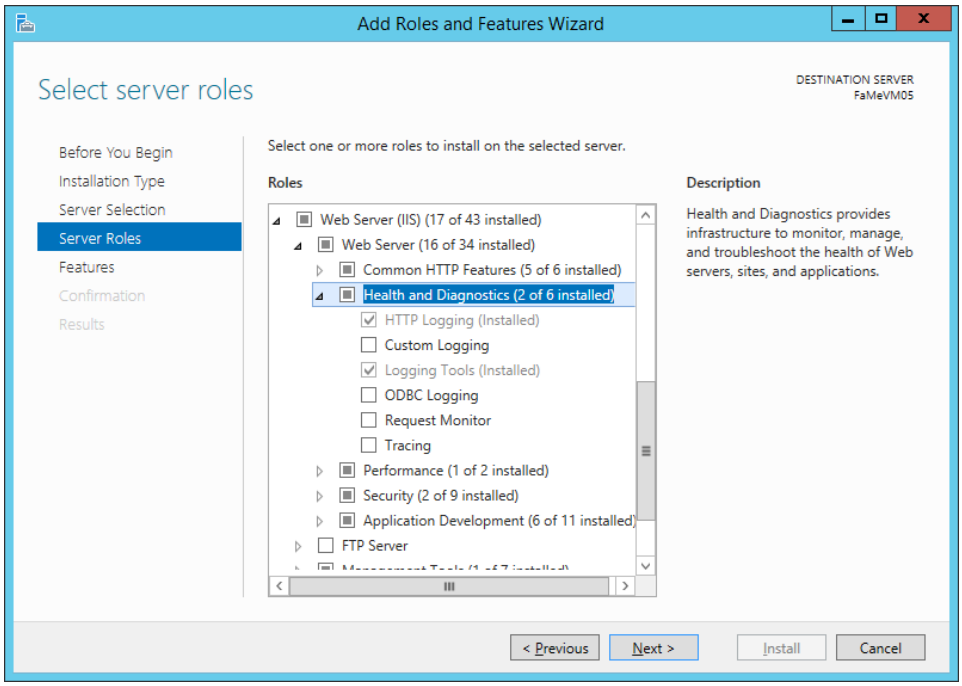


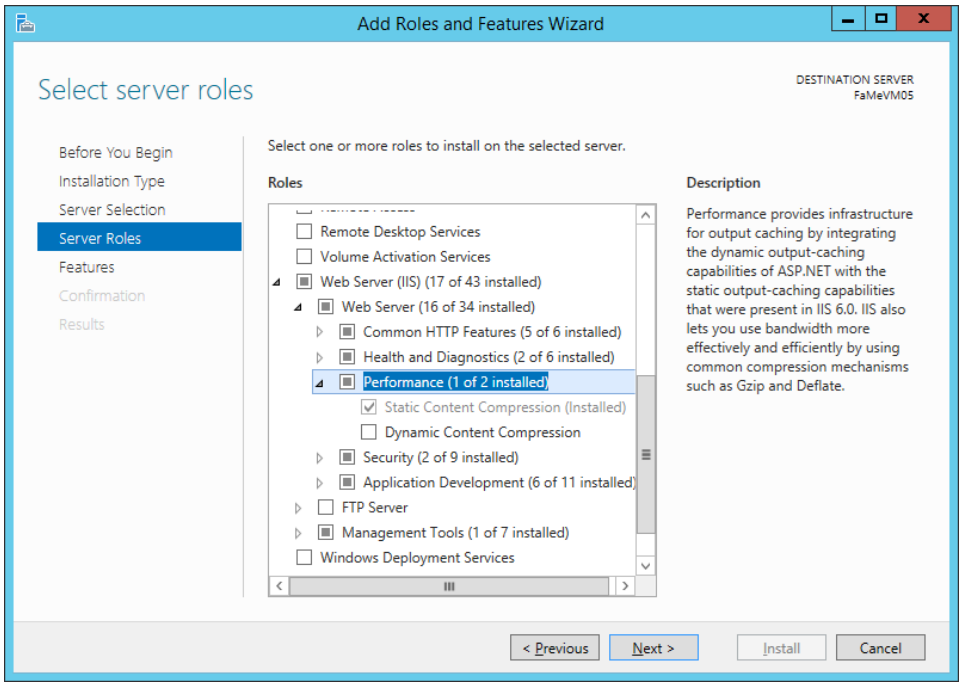
Adding the IIS will ask for the management console to be added as well:

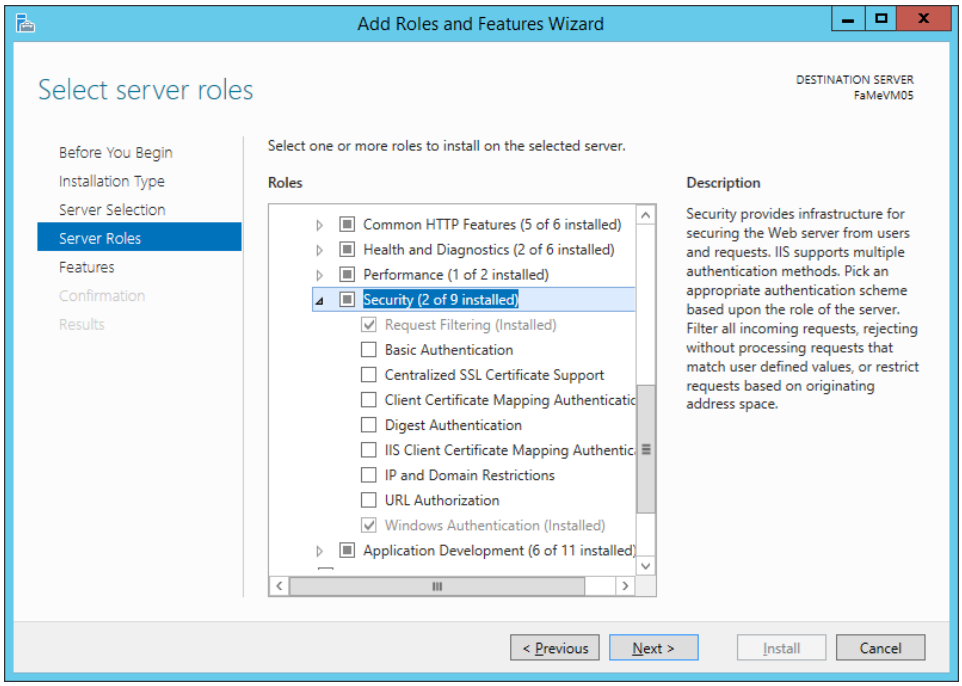


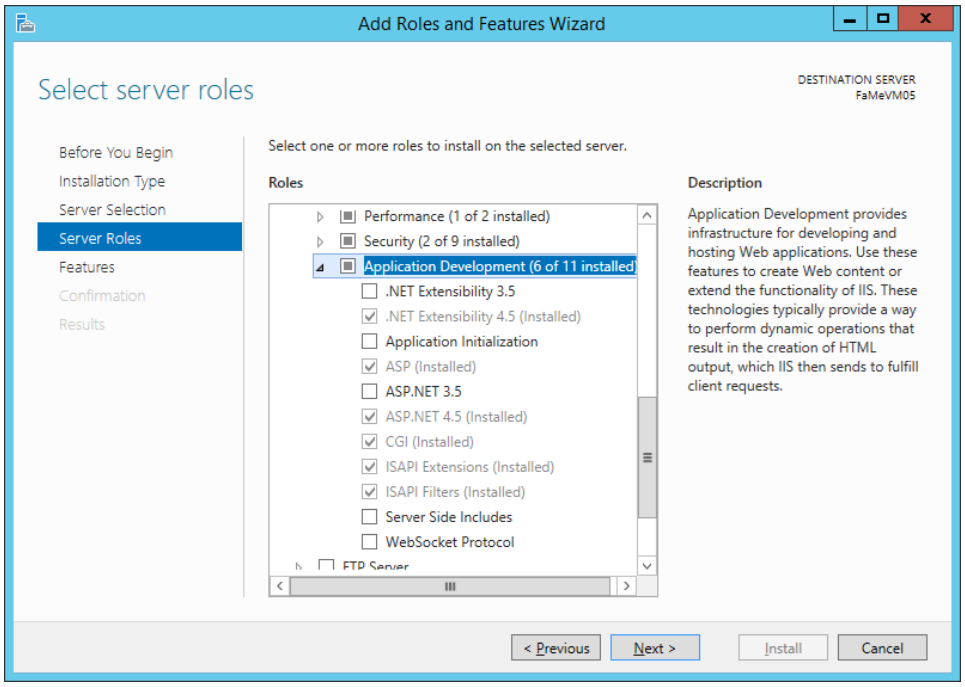
The following screenshots show the IIS options which are either required or recommended for the FaMe IIS plugin:


Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
FaMeVM05
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Web Server (IIS) (17 of 43 installed)
Web Server (16 of 34 installed)
Common HTTP Features (5 of 6 installed)
Default Document (Installed)
Directory Browsing (Installed)
HTTP Errors (Installed)
Static Content (Installed)
HTTP Redirection (Installed)
WebDAV Publishing
Health and Diagnostics (2 of 6 installed)
Performance (1 of 2 installed)
Security (2 of 9 installed)
Application Development (6 of 11 installed)
FTP Server
Description
Security provides infrastructure for securing the Web server from users and requests. IIS supports multiple authentication methods. Pick an appropriate authentication scheme based upon the role of the server. Filter all incoming requests, rejecting without processing requests that match user defined values, or restrict requests based on originating address space.
< Previous
Next >
Install
Cancel

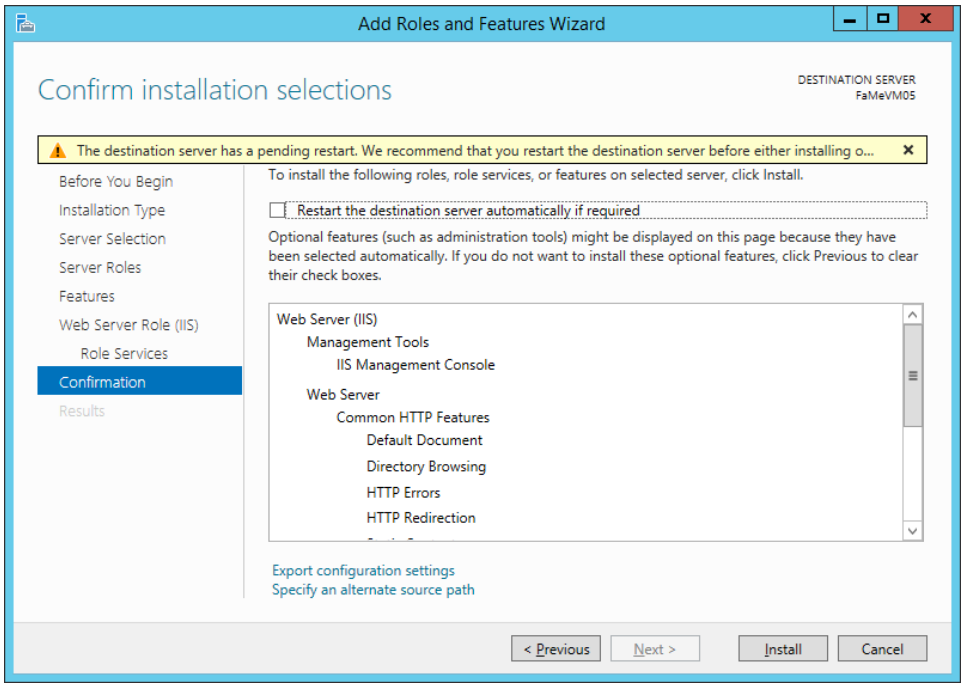

Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
FaMeVM05
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Web Server (IIS) (17 of 43 installed)
Web Server (16 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (2 of 6 installed)
HTTP Logging (Installed)
Custom Logging
Logging Tools (Installed)
ODBC Logging
Request Monitor
Tracing
Performance (1 of 2 installed)
Security (2 of 9 installed)
Application Development (6 of 11 installed)
FTP Server
Description
Health and Diagnostics provides infrastructure to monitor, manage, and troubleshoot the health of Web servers, sites, and applications.
< Previous
Next >
Install
Cancel


Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
FaMeVM05
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Remote Desktop Services
Volume Activation Services
Web Server (IIS) (17 of 43 installed)
Web Server (16 of 34 installed)
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (2 of 6 installed)
Performance (1 of 2 installed)
Static Content Compression (Installed)
Dynamic Content Compression
Security (2 of 9 installed)
Application Development (6 of 11 installed)
FTP Server
Management Tools (1 of 7 installed)
Windows Deployment Services
Description
Performance provides infrastructure for output caching by integrating the dynamic output-caching capabilities of ASP.NET with the static output-caching capabilities that were present in IIS 6.0. IIS also lets you use bandwidth more effectively and efficiently by using common compression mechanisms such as Gzip and Deflate.
< Previous
Next >
Install
Cancel

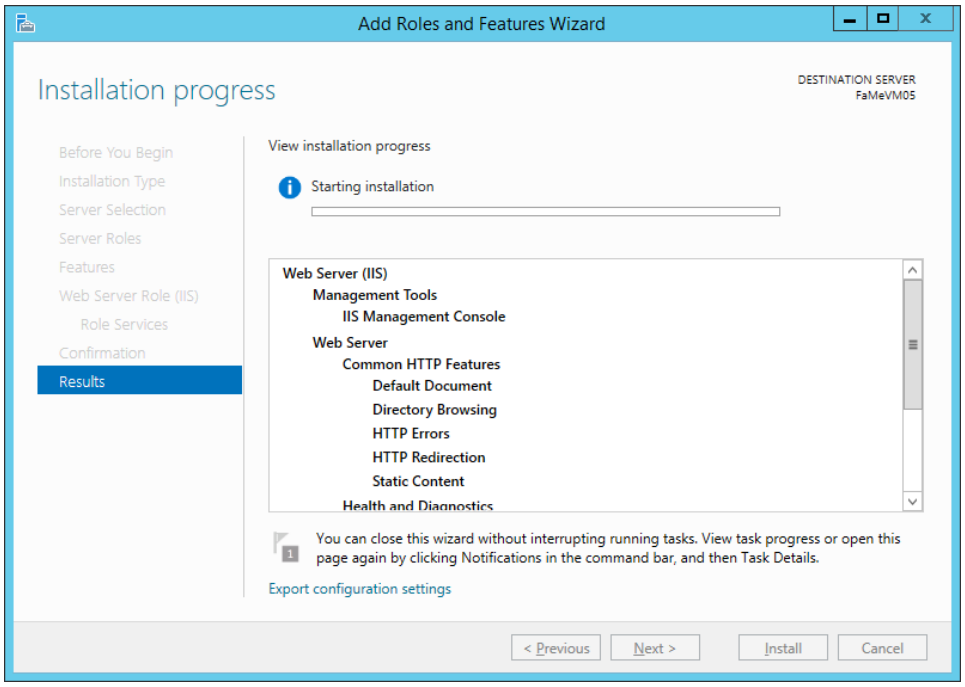

Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
FaMeVM05
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Common HTTP Features (5 of 6 installed)
Health and Diagnostics (2 of 6 installed)
Performance (1 of 2 installed)
Security (2 of 9 installed)
Request Filtering (Installed)
Basic Authentication
Centralized SSL Certificate Support
Client Certificate Mapping Authenticatic
Digest Authentication
IIS Client Certificate Mapping Authentic
IP and Domain Restrictions
URL Authorization
Windows Authentication (Installed)
Application Development (6 of 11 installed)
Description
Security provides infrastructure for securing the Web server from users and requests. IIS supports multiple authentication methods. Pick an appropriate authentication scheme based upon the role of the server. Filter all incoming requests, rejecting without processing requests that match user defined values, or restrict requests based on originating address space.
< Previous
Next >
Install
Cancel



After adding all required IIS features click „Next“ to proceed…



… and finally “Install”:


Add Roles and Features Wizard
Installation progress
DESTINATION SERVER
FaMeVM05
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results
View installation progress
Starting installation
Web Server (IIS)
Management Tools
IIS Management Console
Web Server
Common HTTP Features
Default Document
Directory Browsing
HTTP Errors
HTTP Redirection
Static Content
Health and Diagnostics
You can close this wizard without interrupting running tasks. View task progress or open this page again by clicking Notifications in the command bar, and then Task Details.
Export configuration settings
< Previous
Next >
Install
Cancel

2.4 Oracle client installation

To connect to the database the FaMe IIS plugin requires an Oracle 12c 64 bit client with the ODP.net library. If this is not present on the server it has to be installed. An Oracle database server software installation includes this client, so an extra client needs not to be installed if the server software is in place.

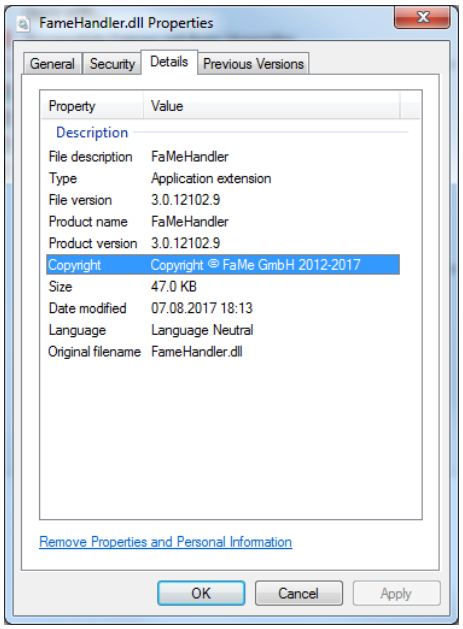
2.4.1 Required Oracle client components

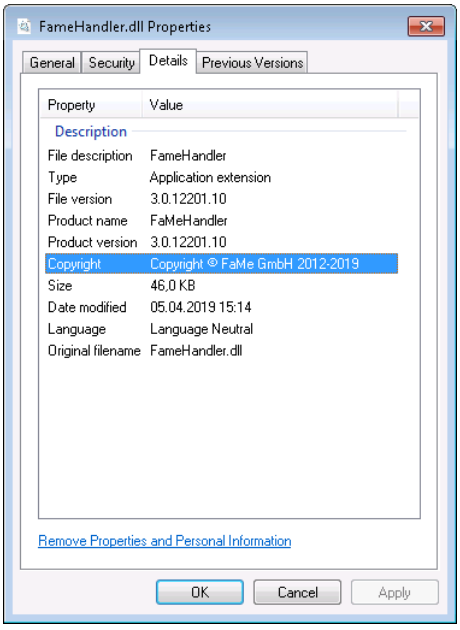
- Oracle Net
- Oracle Data Provider for .NET (ODP.net)
- Oracle Data Provider for ASP.NET
- Optionally / recommended:
- SQL*Plus

The Oracle Data Provider libraries are either selected explicitly on installation or may be installed as part of the *Oracle Programmer* option which can be selected from the component list when running the installation.

2.4.2 FaMe IIS Plugin versions and the required Oracle client software version

The version of the ODP.net library the FaMe IIS plugin has been compiled against is part of the plugin's version number, shown here for both Oracle 12.1.0 and Oracle 12.2.0:





Here, the version of the FaMe IIS plugin for Oracle 12.1 is 3.0.12102.9 and has been compiled against the Oracle ODP.net library version 12.1.0.2, requiring an Oracle 12.1.0.2 client.

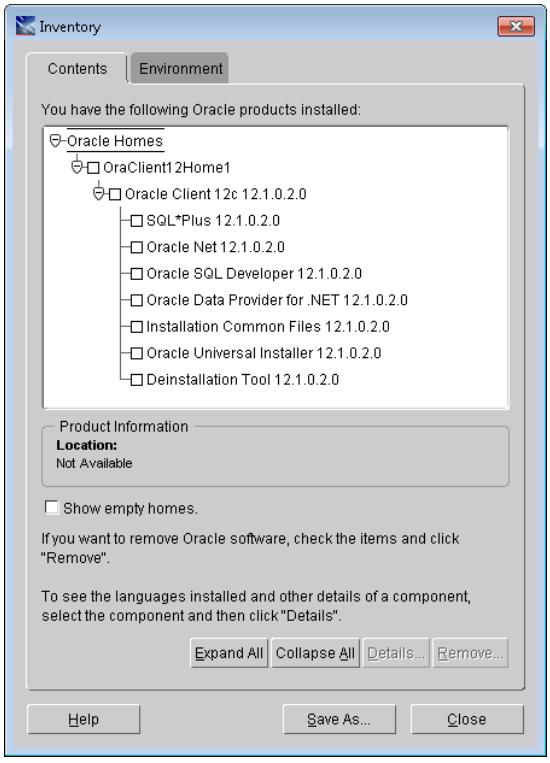
For Oracle 12.2 clients the FaMe IIS plugin's version number is X.X.12201.X. As of April 2019 the current version is 3.0.12201.10.

2.4.3 Verifying installed Oracle client software

To verify the version and components of an already installed Oracle client use the *Universal Installer*. If there are multiple Oracle clients installed on the server be sure to start the one from the appropriate Oracle 12c home.

Oracle Universal Installer (OUI) start screen

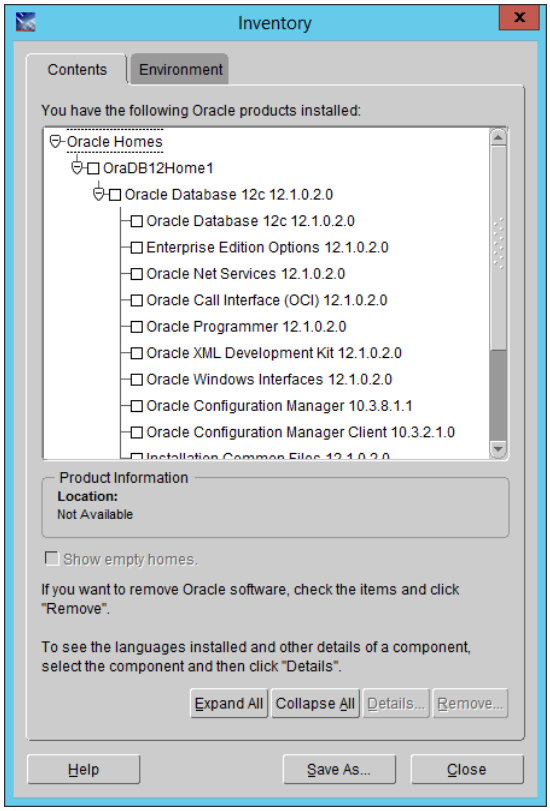
Click *Installed Products* to see what is actually installed:



Here, all required components are installed in version 12.1.0.2. Please note that the *Oracle SQL Developer* is optional and not needed on the server.

2.4.3.1 Oracle client part of an Oracle server installation

If the installed Oracle software is the database server software, the list of installed components will look similar to this:



The *Oracle Data Provider for .NET* is here part of *Oracle Programmer*.

2.4.4 Installing the Oracle 12.1 or Oracle 12.2 client

If the installation of the FaMe IIS Plugin is done directly on the database server no client needs to be installed.

If there is neither an Oracle server software package nor the Oracle client on the machine the client must be installed.

Oracle client software packages are available for download at
https://www.oracle.com/downloads/index.html

2.4.5 ODP.net libraries and their registration in the *Windows Global Assembly Cache* (GAC)

The *Oracle Data Provider for .NET* (ODP) consists of 2 DLLs which are required for the FaMe IIS plugin to run:

Oracle.dataaccess.dll

oracle.web.dll

The Oracle installer puts them into the Windows *Global Assembly Cache* (GAC) in the Windows directory. Standard paths for Oracle 12.1.0.2 are

```
C:\Windows\Microsoft.NET\assembly\GAC_64\
Oracle.Web\v4.0_4.121.2.0__89b483f429c47342\oracle.web.dll

C:\Windows\Microsoft.NET\assembly\GAC_64\
   Oracle.DataAccess\v4.0_4.121.2.0__89b483f429c47342\
```

```
Oracle.dataaccess.dll
```

For Oracle 12.2.0.1 files inside the GAC typically have these paths:

```
C:\Windows\Microsoft.NET\assembly\GAC_MSIL\Oracle.Web\v4.0_4.122.1.0__89b
483f429c47342\oracle.web.dll

C:\Windows\Microsoft.NET\assembly\GAC_64\Oracle.DataAccess\v4.0_4.122.1.0
__89b483f429c47342\ Oracle.dataaccess.dll
```

Here *oracle.web.dll* has moved from 64 bit to MSIL.

If one of these files is not available in the GAC and there is no more than one Oracle client to be installed on the machine, the files can be registered using the *oraprovcfg.exe* tool that comes with the oracle software installation. The Oracle software packages usually do not register the DLLs in the GAC, but for the FaMe IIS Plugin there have to be registered in order to be found by the system when the *FameHandler.dll* is loaded by the IIS.

Example registration of both libraries for an Oracle 12.2.0.1 server:

The Oracle server software is installed in
`C:\oracle\product\12.2.0\dbhome_1`
in this case.

Open a *cmd* prompt window as administrator and cd into the directory containing the *oracle.web.dll* file. It is inside the directory `asp.net\bin\4` that is inside the *ORACLE_HOME* directory on the machine in question:

```
C:\>cd C:\oracle\product\12.2.0\dbhome_1\ASP.NET\bin\4
ASP.NET\bin\4> OraProvCfg.exe /action:gac
providerpath:C:\oracle\product\12.2.0\dbhome_1\ASP.NET\bin\4\Oracle.Web.d
ll

INFO: c:\oracle\product\12.2.0\dbhome_1\asp.net\bin\4\oracle.web.dll is
registered successfully in GAC.
```

Next, do the same for *oracle.dataaccess.dll*:

```
C:\>cd C:\oracle\product\12.2.0\dbhome_1\ODP.NET\bin\4
ODP.NET\bin\4> OraProvCfg.exe /action:gac
/providerpath:C:\oracle\product\12.2.0\dbhome_1\ODP.NET\bin\4\Oracle.Data
Access.dll

INFO:
c:\oracle\product\12.2.0\dbhome_1\odp.net\bin\4\oracle.dataaccess.dll is
registered successfully in GAC.
```

The command to issue don't fit into single lines here and are shown with yellow backgrounds.

2.4.6 TNS configuration for database connections vs. EZCONNECT

Oracle provides 2 basic methods to define database connections:

The *tnsnames.ora* file which may contain multiple connection definitions. All connections are accessible by their names.
It is available to all client programs accessing the database.

The *EZCONNECT* method specifying the 3 basic parameters server, port number, and Oracle instance name as a string. This has to be done everywhere it is used.

2.4.6.1 Using a TNS hostname

The connection to the Oracle database may be specified as a TNS hostname or as an *EZCONNECT* host string. A hostname is a named configuration in the *tnsnames.ora* file which is stored in the following path:

```
%ORACLE_HOME%\network\admin\tnsnames.ora
```

The ORACLE_HOME variable is the path to the Oracle installation directory, for instance

```
C:\app\oracle\product\12.1.0\client_1
```

A hostname setup in the tnsnames.ora file looks like this :

```
FAMEDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = famedbsrv)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = FAMEDB)
    )
  )
```

This example defines a hostname *FAMEDB* for a database service named *FAMEDB* that is running on server *famedbsrv* and is available via port 1521. Please note that instead of a SERVICE_NAME the database has to be addressed by its SID. The database administrator can tell which parameters are to be used here.

In the *web.config* file defining the IIS configuration for the FaMe IIS plugin we can now set the database connection as

```
connect_string="FAMEDB"
```

2.4.6.2 Using EZCONNECT

The EZCONNECT method does not use the tnsnames.ora file but specifies the database connection parameters directly:

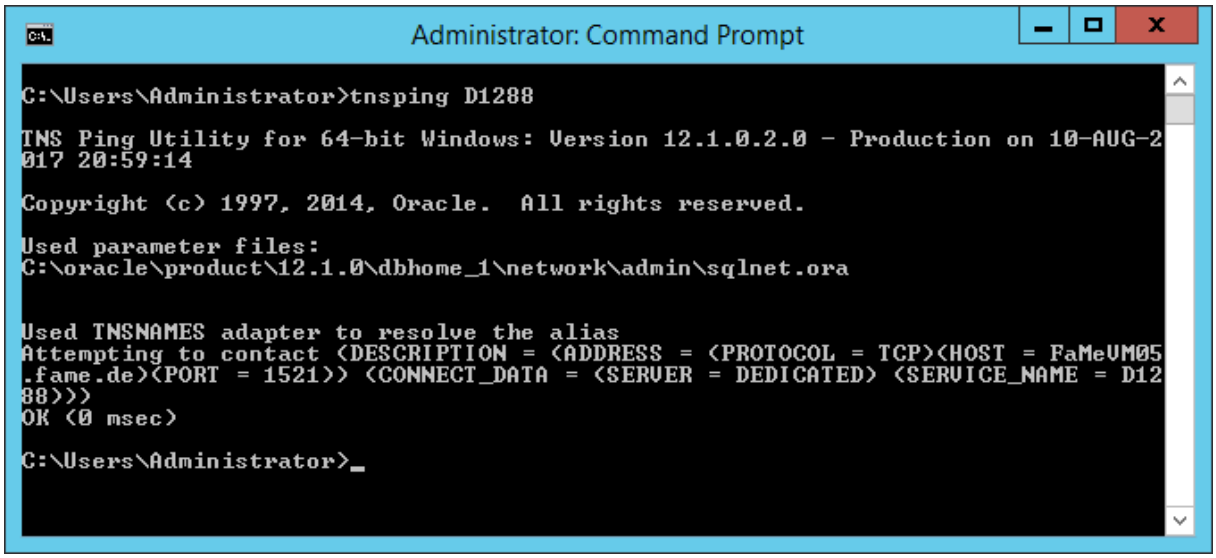
```
connect_string="famedbsrv:1521/FAMEDB"
```

If this way of configuration is used for the FaMe IIS plugin, a tnsnames.ora entry is not required if the FaMe IIS plugin is the only application using an Oracle client on the machine in question.

2.4.7 Testing the Oracle database connection

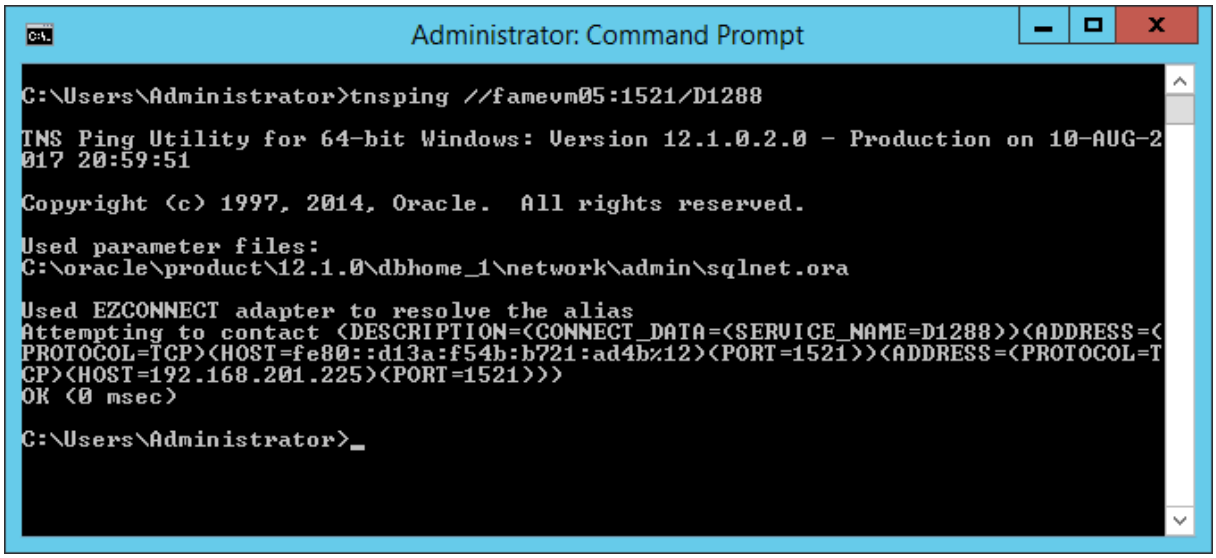
2.4.7.1 Using *tnsping*

To test the database connection there is a command line tool named *tnsping*. It can be used to test connections defined in the tnsnames.ora file :



This test shows that a connection named „D1288" is correctly defined as the database can be pinged. This does not mean, however, that the SERVICE_NAME parameter or the SID_NAME parameter in the *tnsnames.ora* file is set correctly.

The same test for EZCONNECT connections:

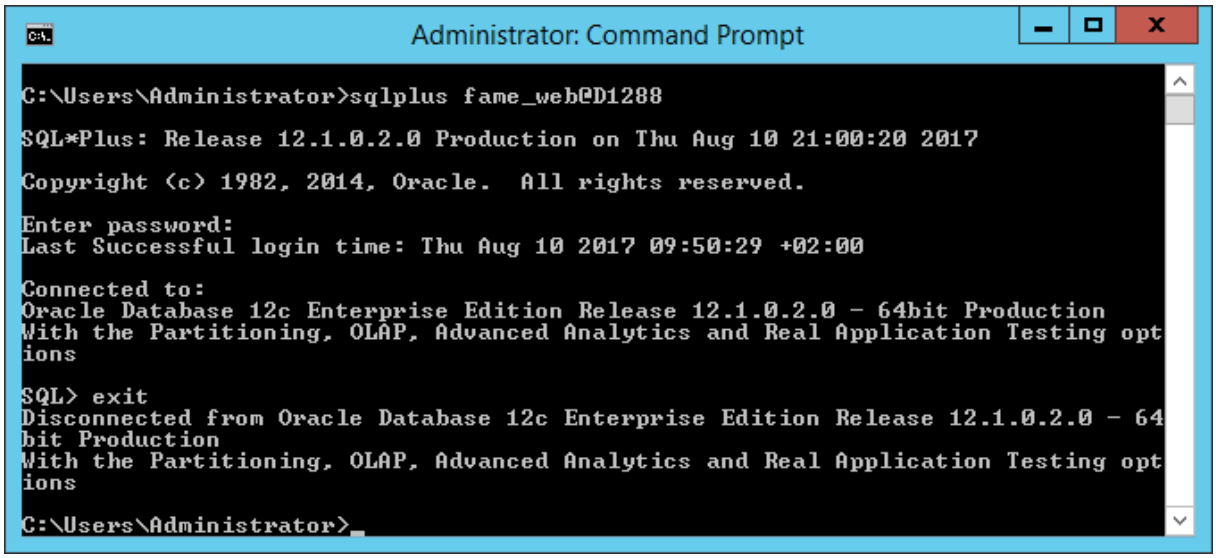


It was tried to ping the server *famevm05* on port 1521. It was successful.

2.4.7.2 Using *SQL*Plus*

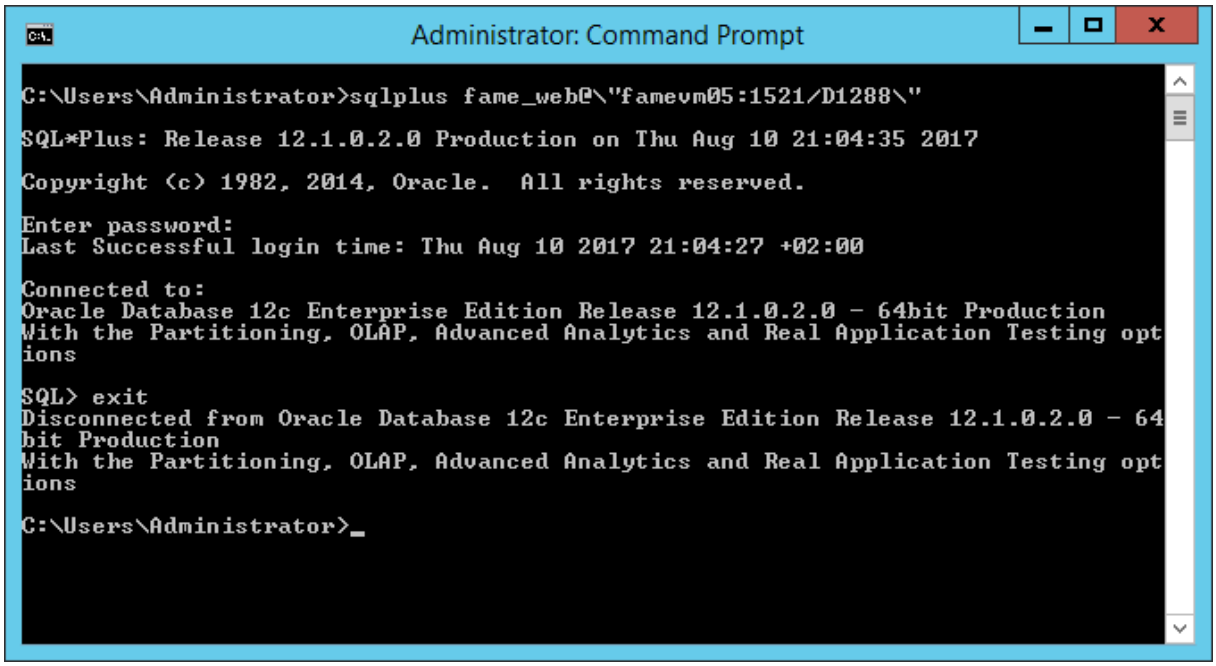
While *tnsping* only allows to ping the database server's listener, SQL*Plus actually connects to the database, verifying that the hostname or EZCONNECT configuration is correct plus verifying that username and password exist.

Connecting with SQL*Plus to test a TNS hostname:



Connecting with SQL*Plus to test a EZCONNECT host string:

Please note that the double quotes are required for SQL*Plus in this case. They have to be escaped by backslashes; otherwise, the command shell will remove them:

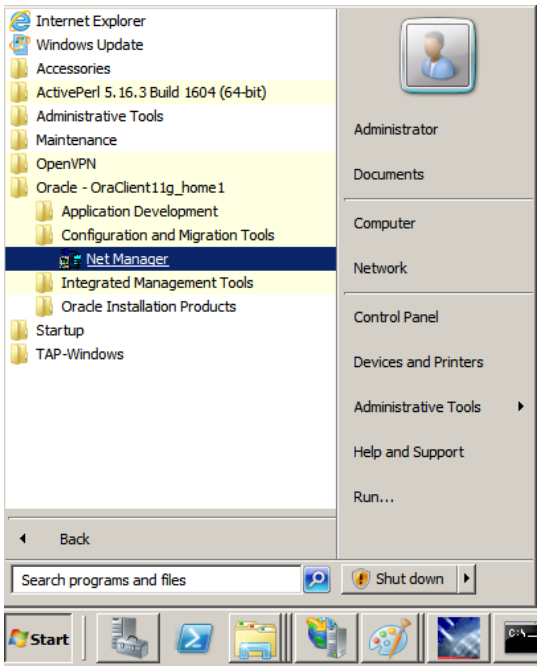


This test was successful, too.

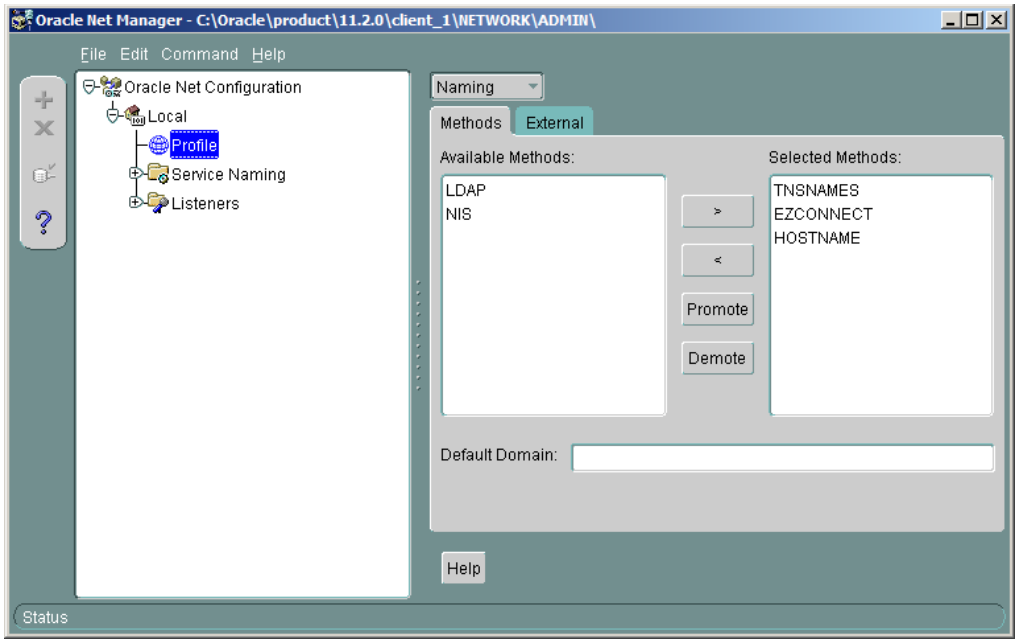
2.4.8 Oracle Net configuration assistant

The Oracle Net configuration assistant can be used to create a first TNS configuration and as well to edit TNS hostnames using a GUI tool. It is recommended to use if there is neither a *tnsnames.ora* nor a *sqlnet.ora* file in the %ORACLE_HOME%\network\admin folder yet.

For a first-time configuration start the Oracle Net Manager:

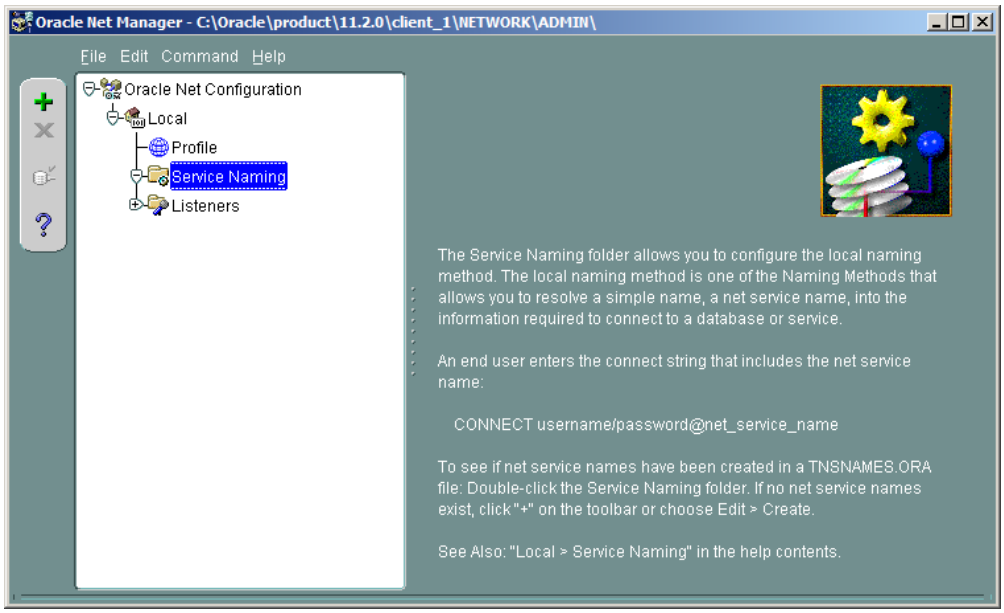


2.4.8.1.1 Oracle Net Manager: common settings



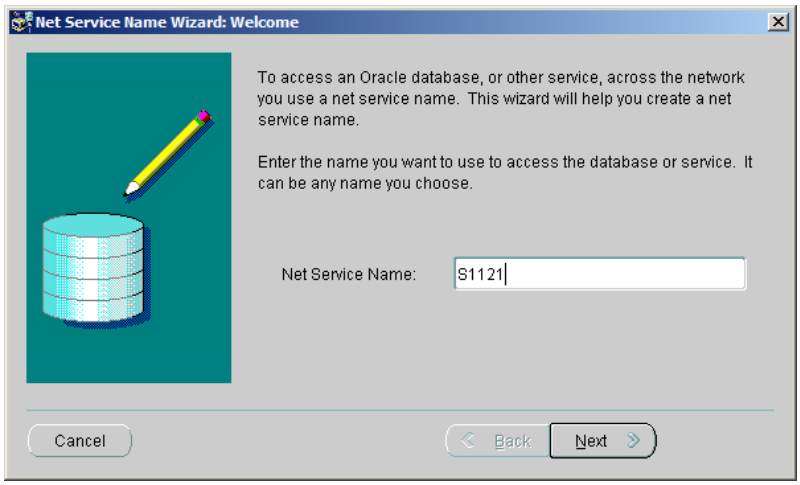
The values shown are the standard values. The required methods are TNSNAMES to configure database hostnames via the *tnsnames.ora* file and EZCONNECT if this method is used to configure the IIS plugin.

2.4.8.1.2 TNS hostname configuration for a database

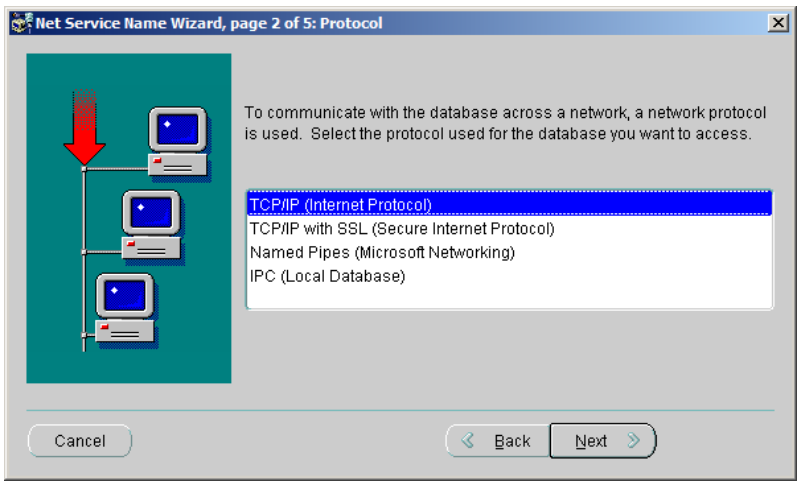


In this dialog window click the  symbol to start the process of adding a new hostname:

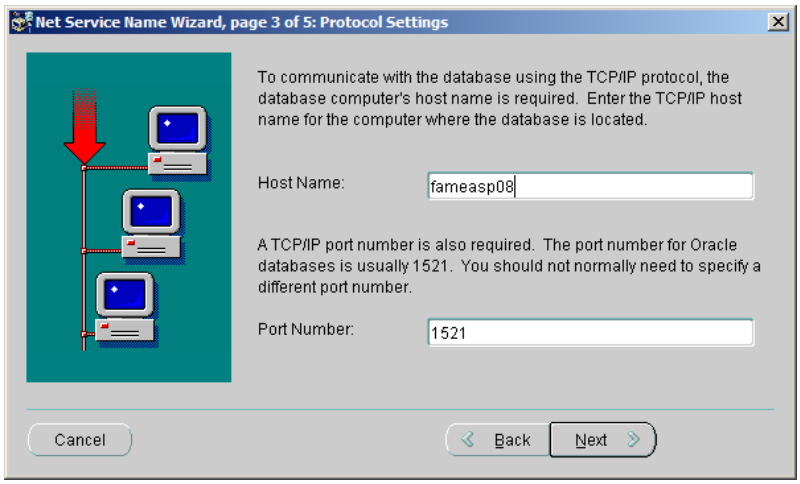
1. Supply the name of the database's hostname here. in this example we used *S1121*:



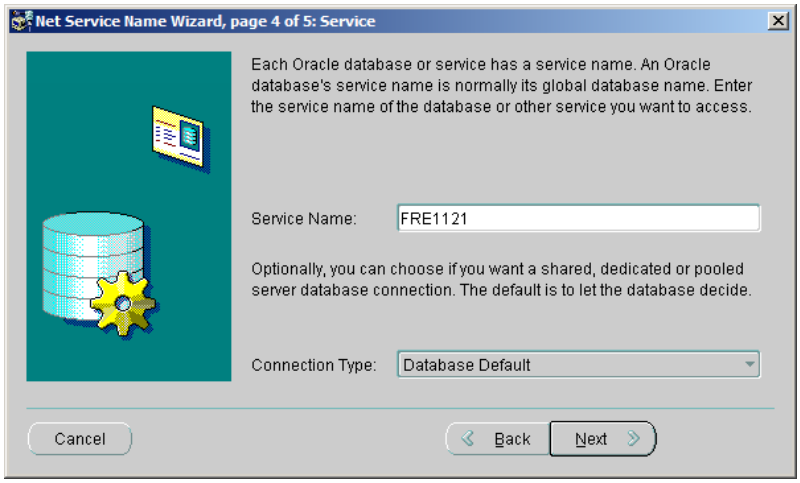
2. The network protocol is TCP/IP:



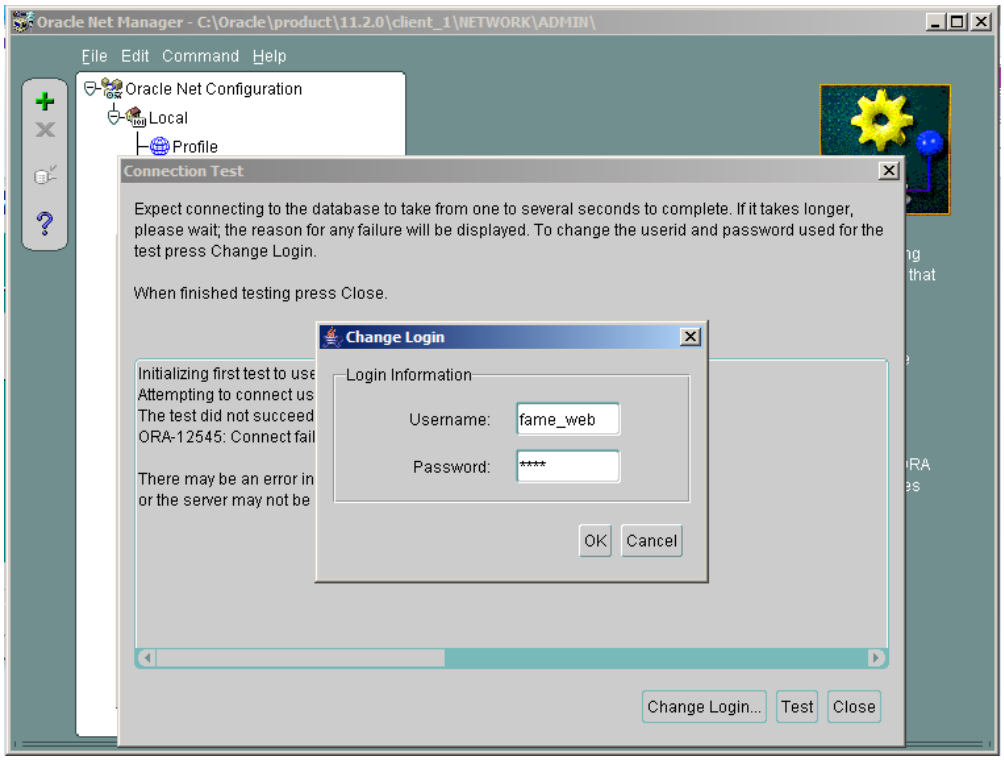
3. Supply the database server's hostname and port number:



4. Supply the database's service name as it is configured inside the *listener.ora* file on the database server:

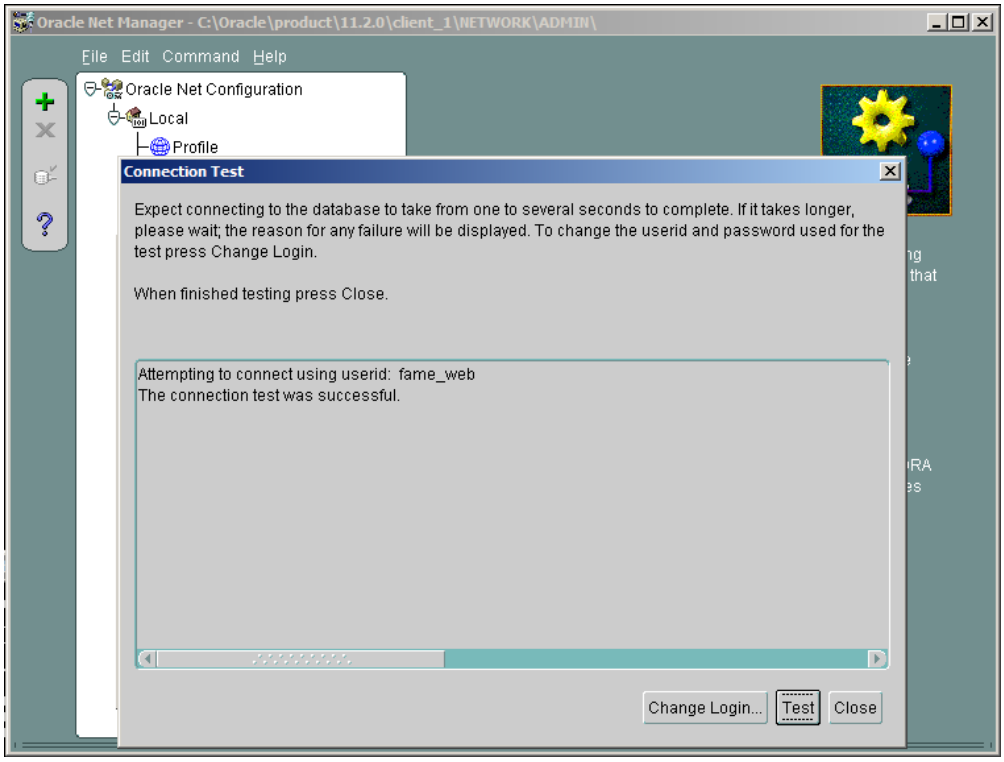


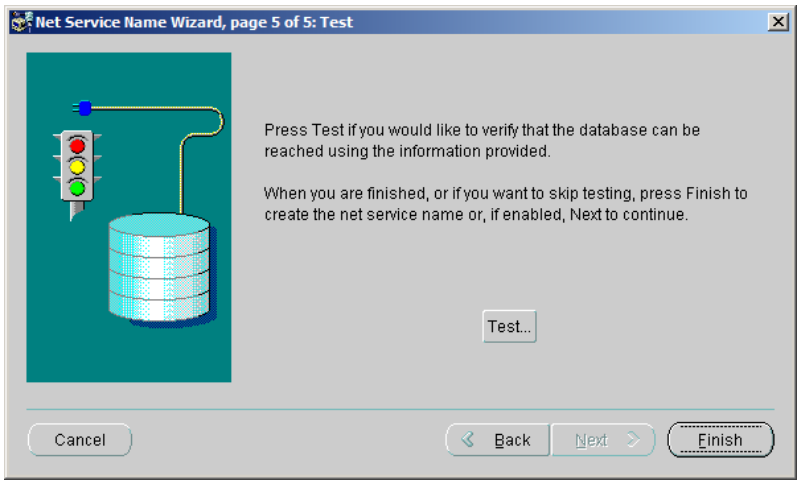
5. Connection test: will be done using the credentials *scott/tiger* automatically. Usually, this account will not exist in the database, and the "Change Login" dialog pops up. Use any existing database account here, for instance the FAME_WEB account:



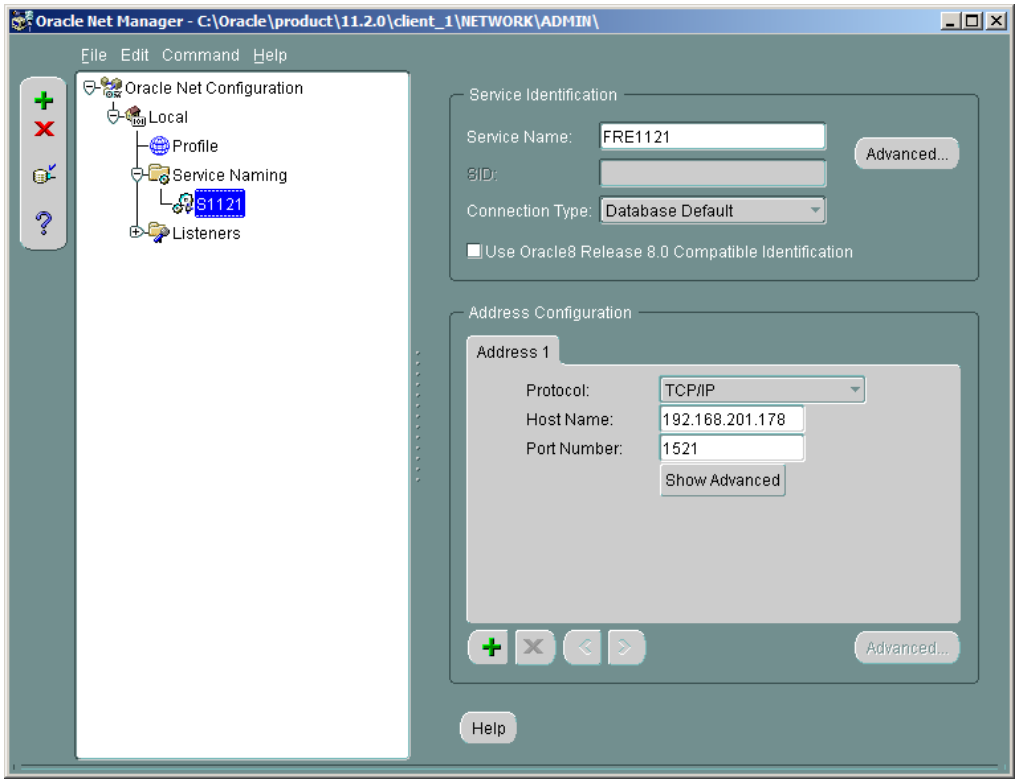
The connection test must work, otherwise check the *listener.ora* file on the database server for SID_NAME oder SERVICE_NAME and the port number. The Windows services *OracleTNSListener* and *OracleService[MyDatabase]* must be up and running for this test.

An Oracle-01017 error basically shows the database is accessible – only the username and/or password used for testing were incorrect.

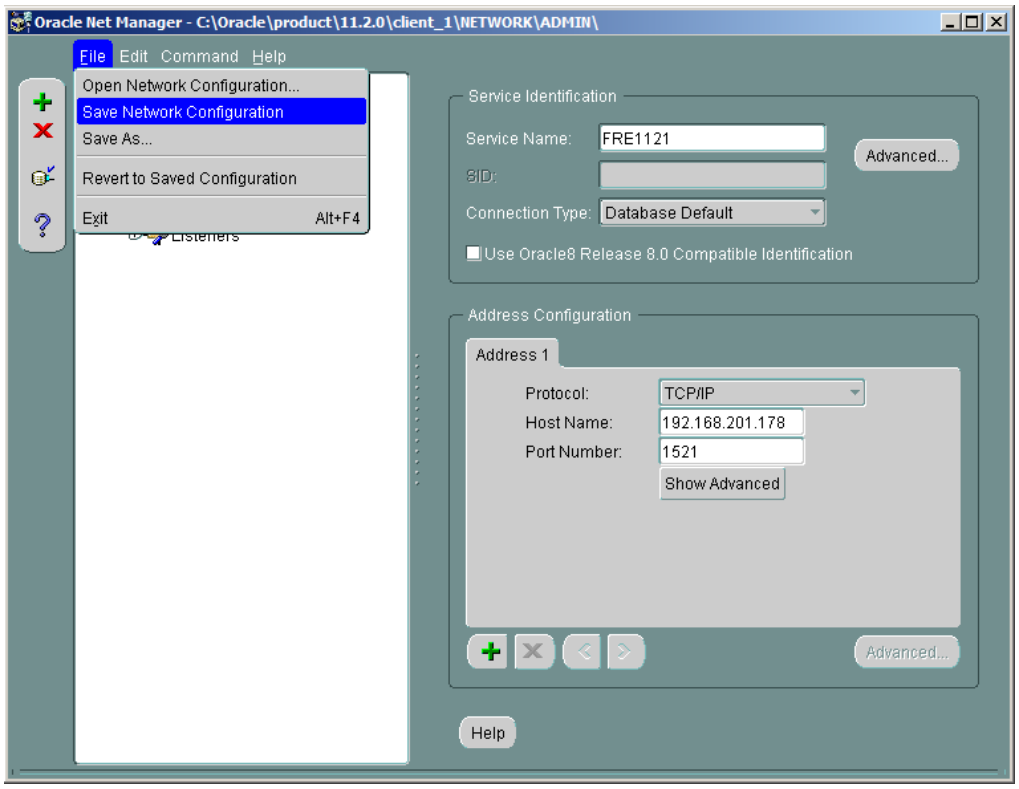




After configuration, the TNS hostname would look like this example when displayed in the Oracle Net manager:



Finally, save the configuration:



The Oracle Net Manager may now be closed.

2.5 IIS configuration for the FaMe IIS Plugin and setup for the first database connection

2.5.1 Overview

Installation and configuration of the FaMe IIS plugin for use with a specific database include these steps:

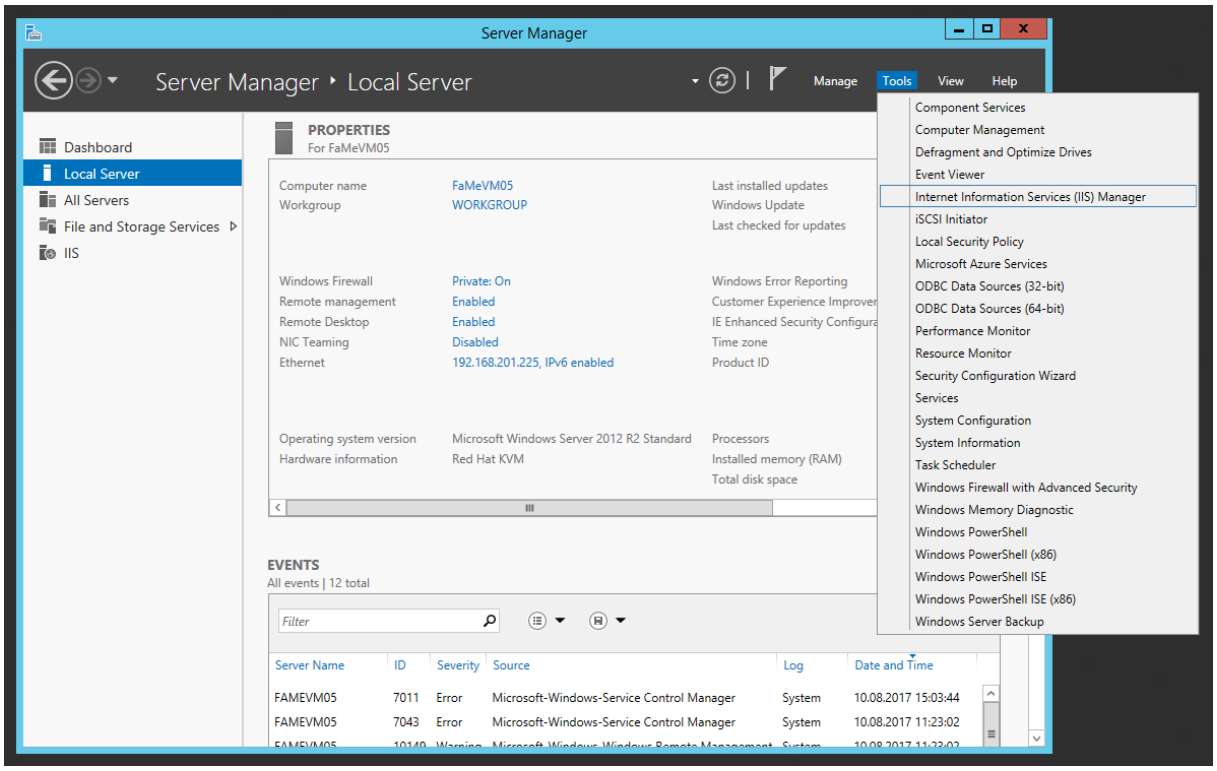
- The file *FameHandler.dll* is to be copied into the *bin* directory directly under the IIS's root directory
- The IIS may have to be enabled to allow for custom handler libraries
- An IIS application pool is to be configured if the default pool shall not be used
- The FaMe application's static files are to be copied into a directory reflecting the application's virtual path
- The static files (HTML, CSS, Javascript) for the FaMe IIS Plugin are to be copied into the application directory
- A `web.config` file is to be prepared defining the database connection and other settings
- If the FaMe PDF is used a Perl interpreter must be installed, together with the FaMe Perl library, the *PDFlib* library for Perl, plus the Perl CGI script and report templates
- The database must be configured for the virtual path to use (COOKIE_PATH setting inside the DEMO_CONFIG table)
- If the Windows single sign-on mechanism is to be used the integrated Windows authentication method must be configured as well.

2.5.2 virtual path examples throughout this document

Most examples in this section use the virtual path `path2app`. In your setup this may be any virtual path; even multiple directory levels are supported.

2.5.3 Launching the IIS manager

The launch the IIS manager use the *Server Manager* → *Tools* → *IIS Manager* as shown:

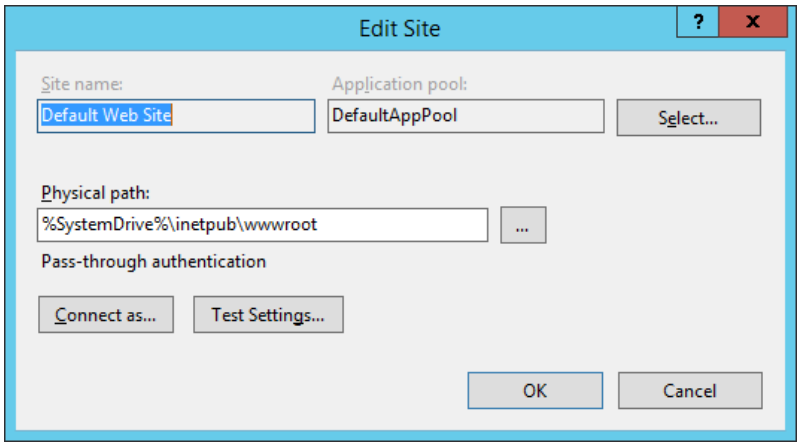


The IIS Manager is also available from *Control Panel* → *Administrative Tools*.

2.5.4 Verification of the site's root directory

The FaMe IIS plugin is going to be installed in a *bin* directory under the site's root directory, and the FaMe application directory is going to be installed there as well.

In IIS manager us the tree to navigate to the site to set up (the default web site in most cases), and click *Basic settings* in the *Actions* panel on the right:

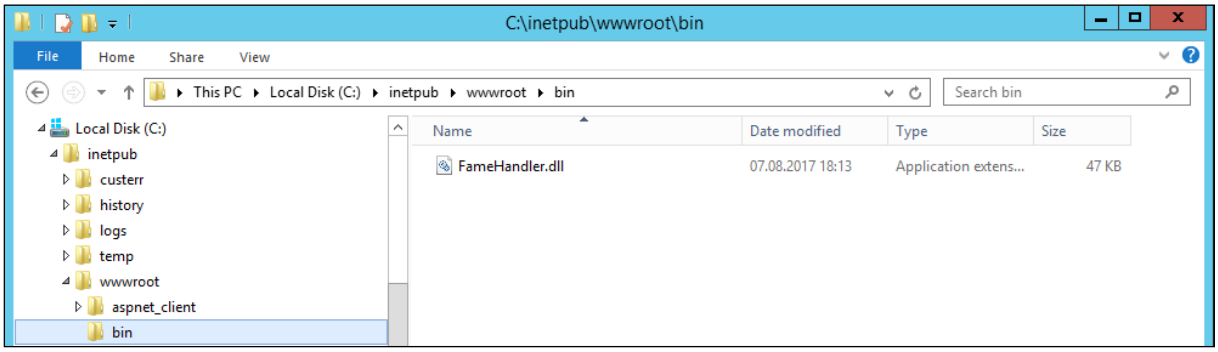


Here the site's root directory is *C:\inetpub\wwwroot*, being the default.

2.5.5 FaMe IIS plugin installation (FameHandler.dll)

2.5.5.1 Copy the DLL into the *bin* directory

The installation of the FaMe IIS plugin requires only to copy the FameHandler.dll file into the *bin* directory under the IIS's webserver root directory.



If the *bin* directory does not exist create it. This will require administrative privileges, as will all further copy and directory actions.

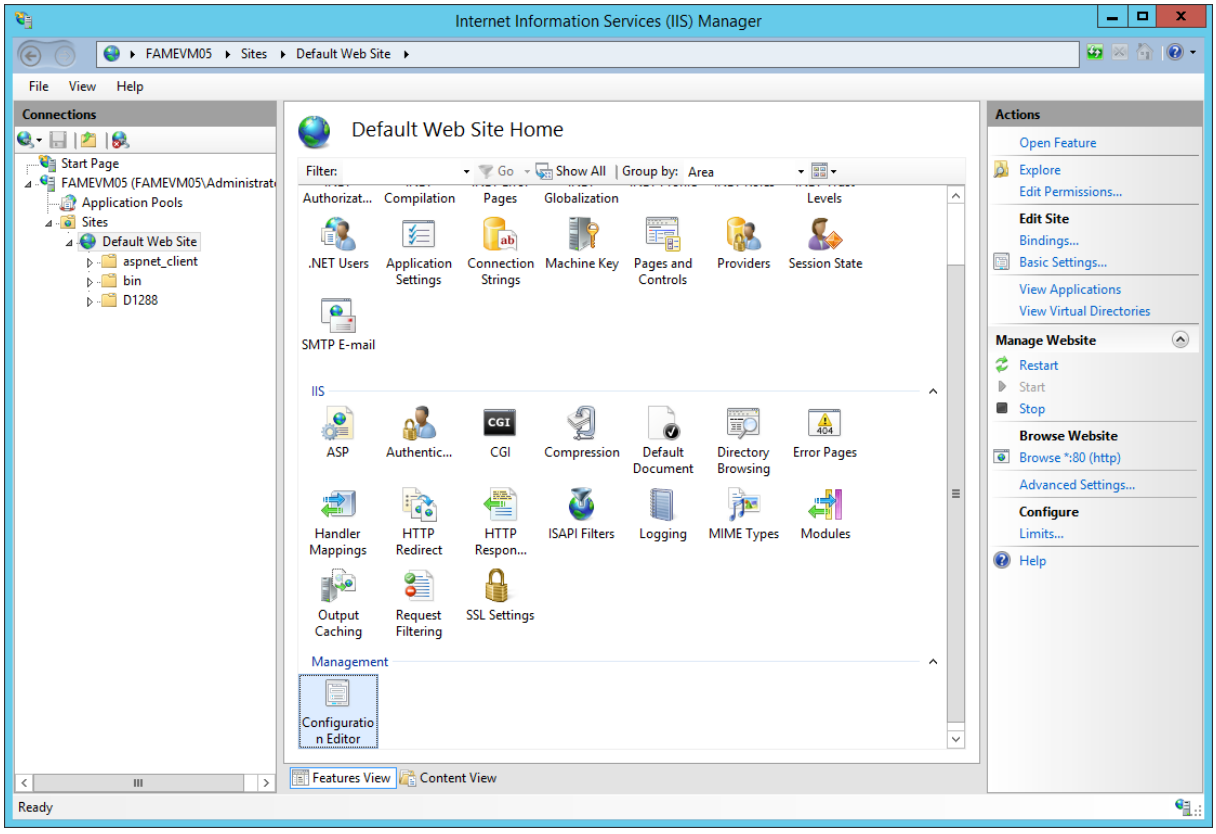
Make sure that the Oracle libraries *oracle.dataaccess.dll* and *oracle.web.dll* are either registered in the Windows Global Assembly Cache (GAC) or copied into the *bin* directory containing *FameHandler.dll*.

See 2.4.5 *ODP.net libraries and their registration in the Windows Global Assembly Cache (GAC)* about the manual registration of the Oracle DLLs in the GAC.

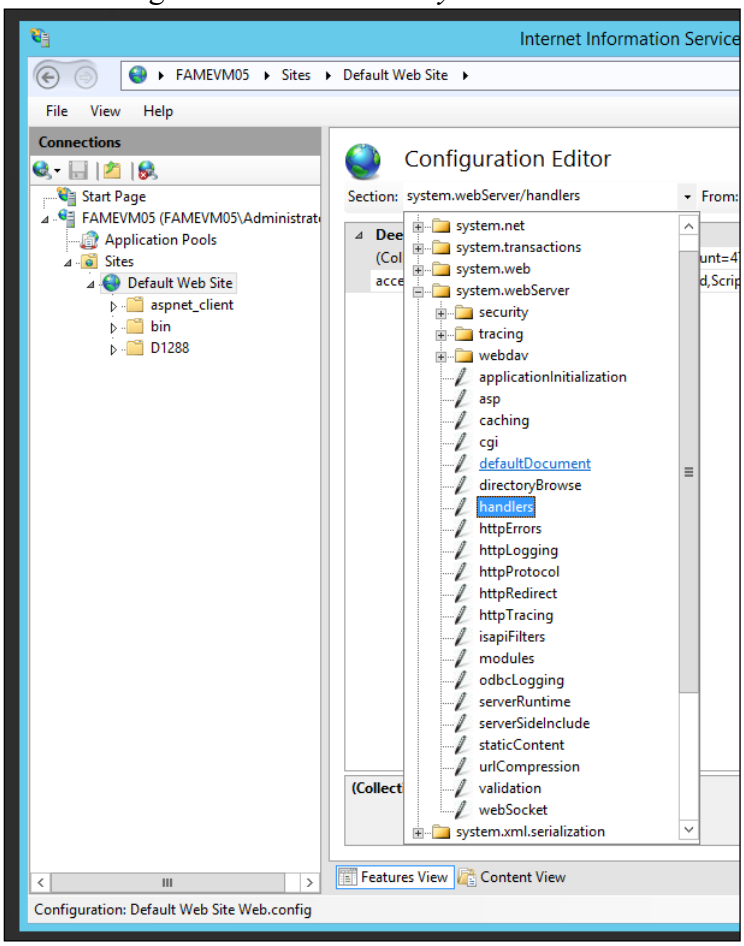
2.5.5.2 Unlock handler mappings

Adding handler mappings is locked by default. The FaMe IIS plugin provides a custom HTTP handler and therefore requires an additional handling mapping.

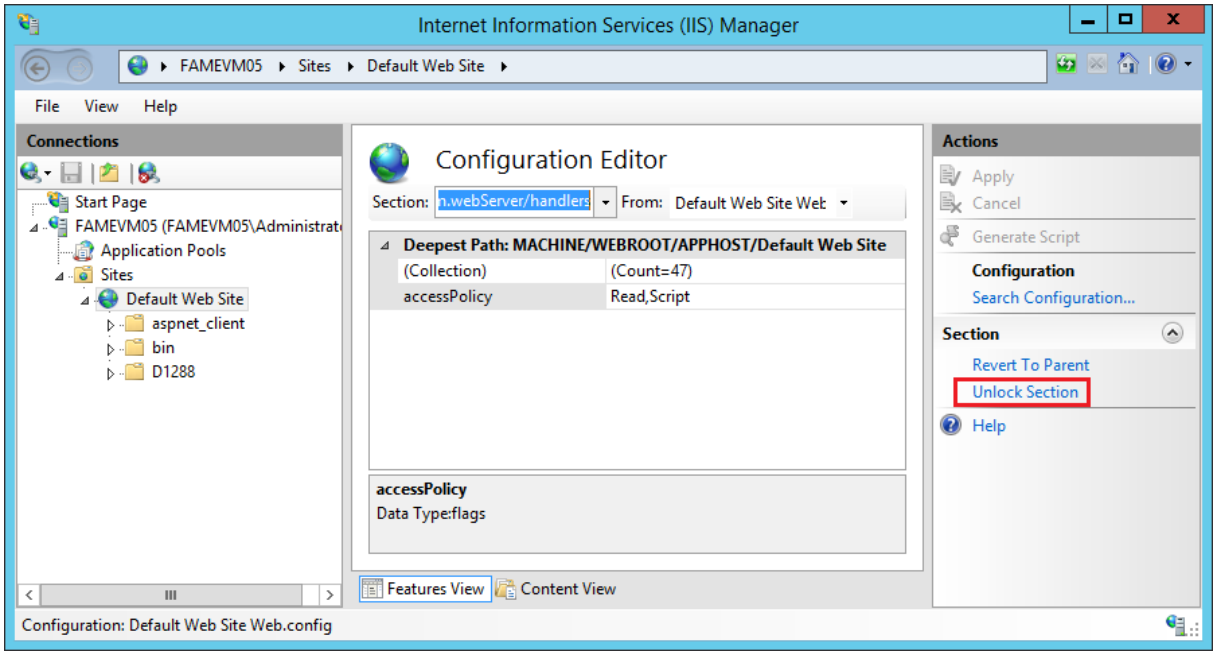
To unlock, open the site's configuration editor:



In the configuration editor select *system.webServer/handlers*:



If the *Action* panel to the right shows a link *Unlock section* click it to unlock. It will show *Lock section* if the unlocking has been done already.

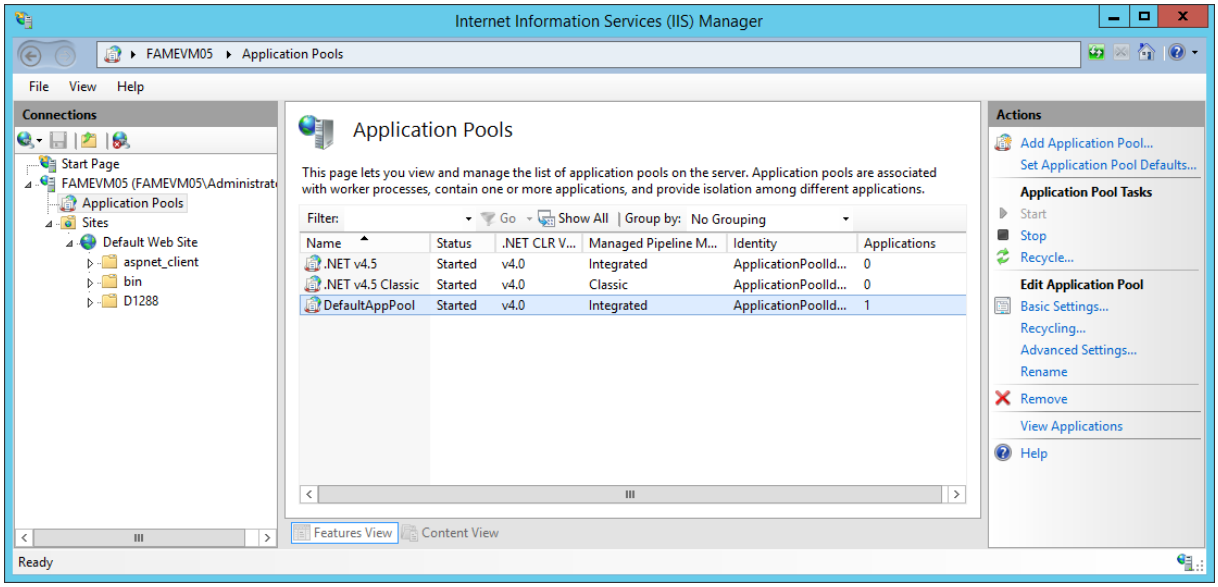


The *accessPolicy* setting *Read,Script* is already the desired setting.

2.5.6 Checking the default application pool or setting up a new one

The IIS 8 default application pool provides the required settings out of the box:

- .net version 4.0
- Integrated pipeline management



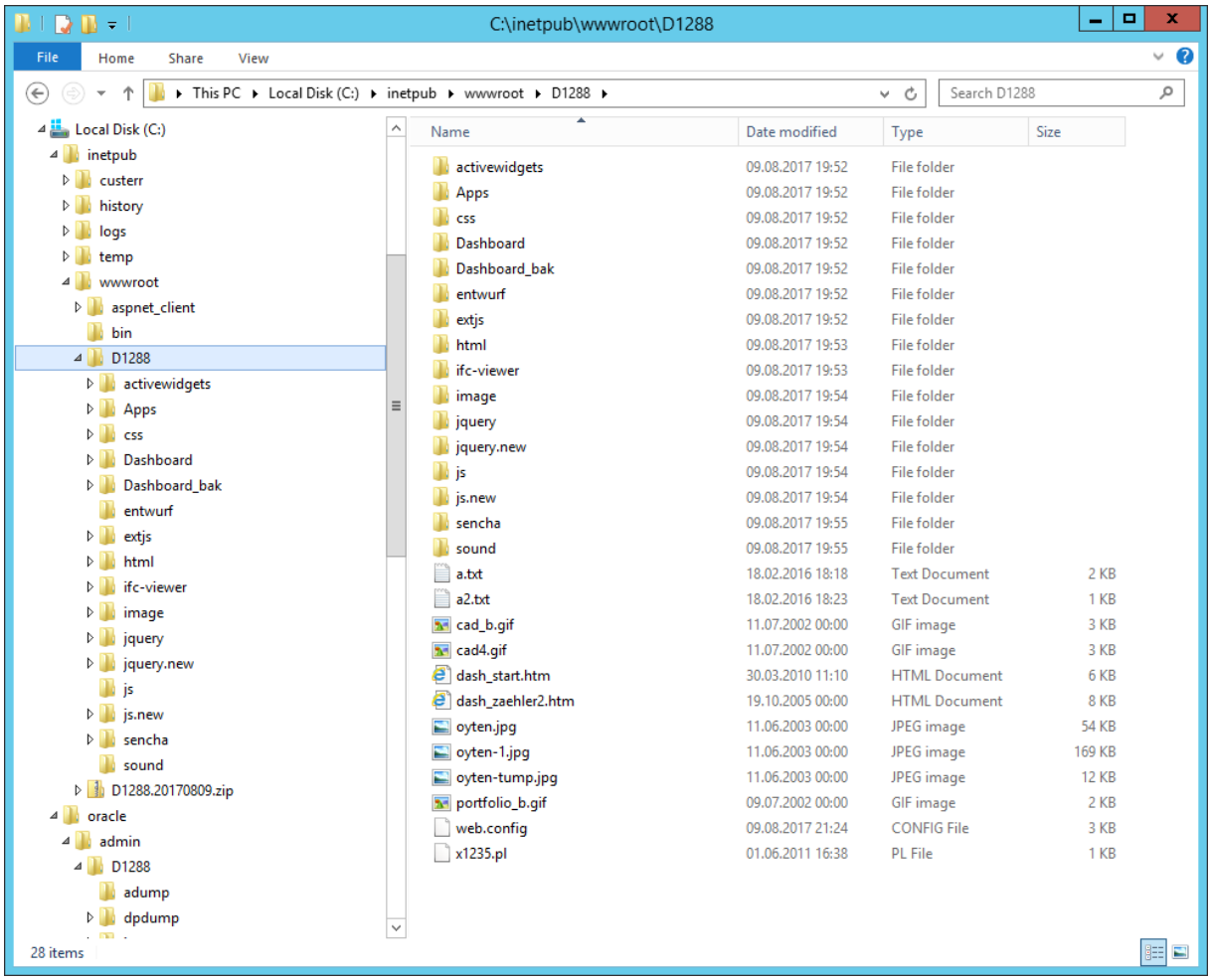
Of course, a separate application pool may be used if desired.

2.5.7 Setting up the FaMe application directory

FaMe FM5 application directories are a set of HTML, image, Javascript, and CSS files. The files are organized in subdirectories *html, js, image, …* and reside all inside the same directory. This superior directory is equivalent to the basic virtual path for the specific database to install. If the path to the application is to be

```
http://server/path2app/
```

then a *path2app* directory must be set up under the site's root. Compare the example:



2.5.7.1 Subdirectories of the FaMe application directory

These subdirectories are typical components of the FaMe application directory:

Path	Type	Description
`html`	Directory	HTML files
`image`	Directory	Image files
`js`	Directory	Javascript and CSS files
`fame`	Virtual path handled by the *FameHandler.dll*	Path to PL/SQL procedures inside the database
`fame/doc`	Virtual path handled by the *FameHandler.dll*	Path to document storage inside the database
`perl`	Script Directory	Contains PDF writer and graphic reporting scripts
`tmpl`	Directory	Contains HTML files with report templates for the PDF writer
`jquery`	Directory	jQuery library folder

`Sencha`	Directory	Extjs library folder

All paths will be addressed relatively (e.g. `../html/<file>`) allowing for arbitrary application paths in this web site.

2.5.7.2 Directory Setup

To set up the directory copy the contents of the file package delivered for your server into a directory equivalent to the virtual path you want to install the application in. If you want to run the FaMe application in

```
http://server/path2app/
```

create a directory path2app under the web site's root and extract the contents of the static file package FaMe provided for your server into it. In case the static file package contains a path itself, i.e. the directories *html, image, js, …* were not in the root of the zip file provided, move these directories into the desired application directory which is *path2app* in this example. See the screenshot above.

2.5.7.3 Adding static files for the FaMe IIS Plugin

The FaMe IIS Plugin requires a few static files to be in place:

Path	Files
`html`	checkDataMismatch.html HttpRequestValidationException.html
`js`	iiserror.css iiserror.js iisversion.css iisversion.js

They are supplied with the plugin installation package and have to be copied in the corresponding directories manually.

2.5.8 web.config file

Copy the *web.config* file provided with the installation package into the application directory or into the site's root directory. Use the root directory only if you are going to configure more than one FaMe application database in this site.

2.5.8.1 Location for the web.config file

The web.config file may be used to configure the IIS for several databases at once. In this case it will contain several entries

```
<add name="FameHandler ../>
```

and it may be located in the document root directory.

If it is preferred to put the web.config file into the application directory the internal setup won't change at all, but it will only be effective for the virtual path mapped to the application directory. If the file is put into `{document_root}\path2app`, which will typically be `c:\inetpub\wwwroot\path2app`, it should not define any *FameHandler* for other virtual paths than `/path2app` and possibly `/path2apppdf` if /path2apppdf is used at all (refer to *2.5.8.4 Second FameHandler definition for the FaMe PDF Writer* for that).

2.5.8.2 Web.config setup

A typical web.config file has this setup:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <configSections>
    <section name="FameConfig" type="FameHandler.FameConfigSection" />
  </configSections>
  <system.web>
    <customErrors mode="Off" />
    <httpRuntime maxRequestLength="40000" requestValidationMode="2.0"/>
  </system.web>
  <system.webServer>
    <handlers accessPolicy="Read, Script">
      <add name="FameHandlerpath2app" path="/path2app/fame"
verb="GET,POST" type="FameHandler.FameHandler" resourceType="Unspecified"
preCondition="integratedMode" />
      <add name="FameHandlerpath2apppdf" path="/path2apppdf/fame"
verb="GET,POST" type="FameHandler.FameHandler" resourceType="Unspecified"
preCondition="integratedMode" />
      <add name="Perl" path="*.pl" verb="GET,POST" modules="CgiModule"
scriptProcessor="C:\ActiveState5.12Perl\bin\perl.exe -
IC:\FaMe\FaMe_FM5_PDF_Writer_2.4.2.2\lib -
IC:\FaMe\perlchartdirector\ChartDirector\lib &quot;%s&quot; %s"
resourceType="File" requireAccess="Script" />
    </handlers>
    <httpErrors existingResponse="PassThrough"/>
    <defaultDocument>
      <files>
        <clear />
        <add value="index.html" />
        <add value="index.htm" />
      </files>
    </defaultDocument>
    <security>
      <requestFiltering allowDoubleEscaping="true">
        <requestLimits maxAllowedContentLength="40000000"
          maxQueryString="4096" />
      </requestFiltering>
    </security>
    <httpProtocol>
      <customHeaders>
        <remove name="X-Powered-By" />
      </customHeaders>
    </httpProtocol>
  </system.webServer>
  <FameConfig>
    <DADCollection>
      <add DAD="path2app"
        connect_string="famevm05:1521/FAMEDB"
        username="fame_web"
        password="fame"
        document_table="fm_document_upload"
        document_path="doc"
        document_proc="document_pak.download_doc"
        before_proc="fame_session_init.set_fame_session_context;"
        after_proc=""
        fame_debug_mode="user"
        fame_debug_path="c:\Logs\path2app"
        fame_character_set=""
        NLS_LANG="GERMAN_GERMANY.AL32UTF8"
        fame_auth="owa_custom_authenticate;"
```

```
        fame_statistic="off"
        min_pool_size="10"
        max_pool_size="100"
        htbuf_len="63"
        result_page_array_size="2000"
        error_type="debug"
        version_page="full"
      />
    </DADCollection>
  </FameConfig>
</configuration>
```

The parts outlined in yellow configure the FaMe IIS plugin, and the parts outlined in green are for the FaMe PDF Writer. Please note that the virtual and physical paths shown here are all examples and my differ on your system.

2.5.8.3 Import settings in the web.config file

Some parameters need to be set in the web.config file for the FaMe application:

Setting	Description
requestValidationMode="2.0"	Required setting to allow for search in FaMe forms using greater-than > and less-than < characters
allowDoubleEscaping="true"	Required to allow for plus signs used in FaMe forms
maxAllowedContentLength = 40000000	Limits the size of uploaded files in BYTES; defaults to 30.000.000. May be set higher or lower as needed
maxRequestLength=40000	Limits the size of uploaded files on KILOBYTES; must be set similar to *maxAllowedContentLength*.
maxQueryString=4096	Max. length of the query string; defaults to 2048

2.5.8.4 Second *FameHandler* definition for the FaMe PDF Writer

If the FaMe PDF Writer is in use and Integrated Windows Authentication is in use too for the FaMe application, a second virtual path mapped to the database is required to enable database connections from the PDF Writer.

The example above shows actually 2 handlers added:

- FameHandlerpath2app => virtual path `/path2app/fame`
- FameHandlerpath2apppdf => virtual path `/path2apppdf/fame`

These are 2 handlers for 2 separate virtual paths that are both mapped to the database.

The second virtual with the *pdf* supplement path is added to enable separate configurations for database access from browsers and from the FaMe PDF Writer.

The reason for this setup is that the FaMe PDF Writer which is a Perl CGI script does not support this Windows authentication and requires anonymous access.

If this setup is used the virtual directory `/path2app` must be configured in the IIS manager for Integrated Windows Authentication and the virtual directory `/path2apppdf` must be

configured for anonymous access. An empty directory `{document_root}\path2apppdf` may be created in the file system to set up the anonymous access schema for this virtual path. If the virtual path `/application/fame` (in the example: `/path2app/fame`) allows for anonymous access this setup is not required.

2.5.8.5 <handlers> and <FameConfig> sections

The `<handlers>` section defines HTTP request handlers to add. Here, the `FameHandler` is to be added which is provided by the FameHandler.dll library that has been installed in the `bin` directory.

`<handlers>` : can contain multiple `<add>` tags to define virtual paths to be processed by the FameHandler.dll

For each virtual path (`/path2app` in this example) that is to be mapped to a database an `<add>` element must be added to the `<handlers>` section:

```
<add name="FameHandlerpath2app" path="/path2app/fame" verb="GET,POST"
type="FameHandler.FameHandler" resourceType="Unspecified"
preCondition="integratedMode" />
```

If a 2nd FameHandler is needed for the PDF writer as described in the previous section the following lines have to be added:

```
<add name="FameHandlerpath2apppdf" path="/path2apppdf/fame"
verb="GET,POST" type="FameHandler.FameHandler" resourceType="Unspecified"
preCondition="integratedMode" />
```

The name attribute of the `<add>` element must start with "`FameHandler`" and be followed by a string that is declared as `DADname` in the `<DADCollection>` element.
For

```
<add name="FameHandlerpath2app" … >
```

there must be a

```
<add DAD="path2app" … >
```

element. The DAD (database access descriptor) defines the database and HTTP parameters for this connection while the handler definition only defines the HTTP handler for the virtual path.

For the

```
<add name="FameHandlerpath2apppdf" … >
```

a section

```
<add DAD="path2apppdf" … >
```

is needed. It will use the same parameters as the `<add DAD="path2app">` element.

Attributes of the `<add>` element:

Parameter	Value	Description
name	FameHandler[DADname]	The name must be "FameHandler" followed by a name declared as DAD="[name]" in the <DADCollection>

path	The virtual path to handle: the application path plus "/fame"	Defines the virtual path mapped to database procedures
Verb	GET.POST	The HTTP verbs to handle
type	FameHandler.FameHandler	Defined by the DLL; constant value
preCondition	integratedMode	Constant definition

`<FameConfig>` contains a `<DADCollection>` element which can contain multiple `<add>` tags.

Parameter	Value	Description
DAD	[DADname]	The DADname from the name parameter of the <handler> that was added above

connect_string	TNS hostname or EZCONNECT host string	The Oracle connection's hostname or parameters (EZCONNECT).
		Basic format:
		`Server:Port/Instance`
		Example:
		`dbserver:1521/TESTDB`

username	"fame_web"	Constant
password	(The password)	fame_web's database password
document_table	The table used to store documents in the database	Always use `document_table= "FM_DOCUMENT_UPLOAD"`
document_path	The subdirectory below fame that contains the document stored in the database	Always use `document_path="doc"`
document_proc	The PL/SQL procedure processing requests to download documents from the database	Default value: `document_proc= "document_pak. download_doc"`
before_proc	The PL/SQL procedure that is to be called before the procedure that is specified by the requests URL	Use `before_proc= "fame_session_init. set_fame_session_context;"` This procedure sets up the context with the session data.
after_proc	The PL/SQL procedure that is to be called before the procedure that is specified by the requests URL	Not used in production environments but may be used for tracing or debug purposes

fame_debug_mode	|off|on|user	`off`: no debug log (production) `on`: each request writes a short diagnostic file into the debug directory `user`: detailed debug log **caution:** large files! The files aggregate inside the log directory – cleanup must be provided separately!
fame_debug_path	A path on disk	The directory debug log files are written into Must be writable by all IIS users Should be individual for each DAD declared
fame_character_set	The character set used to encode HTTP data	Sets the character set to assume for HTTP requests and to use for HTTP responses. Overrides NLS_LANG Default: empty – use NLS_LANG instead
NLS_LANG	An Oracle NLS string	Client-side language, territory, and character set settings for the Oracle client. Effects the encoding of HTTP requests and reponses as well.
fame_auth	Authentication check procedure	Name of custom PL/SQL procedure to verify client authentication before processing the request. Currently not implemented.
fame_statistic	Reserved parameter	Currently not in use – leave empty.
min_pool_size	Minimum pool size for Oracle connection pool	The minimum number of Oracle sessions held open by the FaMe IIS Plugin. Valid values: 1-100 Default value: 5
max_pool_size	Maximum pool size for Oracle connection pool	The maximum number of Oracle sessions held open by the FaMe IIS Plugin. Valid values: 10-1000 Default value: min_pool_size * 2
htbuf_len	PL/SQL toolkit line buffer length, in characters	Valid values: 30-63 for Unicode databases 30-255 for 8-Bit encodings Always use `htbuf_len=63` for Unicode databases (AL32UTF8) and `htbuf_len=255` for 8-Bit databases,

		i.e. for character sets like WE8MSWIN1252, US7ASCII, etc.
result_page_array_size	Number of text lines retrieved from database per network roundtrip	Default = 1000 May be increased for performance This is *not* the number of rows retrieved by database queries, but the number of text lines retrieved from the PL/SQL toolkit's internal text buffer per network roundtrip from the FaMe IIS plugin to the database
error_type	The type of error page produced on database errors	Valid values: 404 = respond HTTP-404 on error 500 = respond HTTP-500 on error "debug" or empty value = print the error message returned by the database Default = empty
version_page	Controls the type of version information page	OFF|DISABLED = no version page; returns HTTP-404 FULL = full version information with configuration details. Please note that this version information is visible to everyone, not just privileged users. empty = short version information

use_procedure_filter	Controls the use of the procedure filter: on or off	By default, procedures matching one of these patterns will be blocked:
		`^SYS.*`
		`^SYSTEM.*`
		`^DBMS_*`
		`^UTL_*`
		`^OWA_*`
		`^OWA.*`
		`^HTF.*`
		`^HTP.*`
		^ = beginning of procedure name
		All procedure names must follow Oracle naming rules. Requests with procedure names not following naming rules will be blocked.
		Requests to blocked procedures result in an HTTP-404 response.
		Setting `use_procedure_filter=no` disables this function. Do not use that in production.

max_logfile_age	The maximum age of logfiles after which a file is to be deleted by cleanup operation	Specify <number>[dhm] to specify a max. age of <n>days, hours, or minutes. Default unit = minutes Default value = 1d
cleanup_log_interval	The interval after which a new logfile cleanup is started.	Specify <number>[dhm] If the *cleanup* file inside the log directory is missing or older than this value old logfiles in this directory are deleted. Default unit = minutes Default value = 1d
max_logfile_count	The maximum number of logfiles kept in the log directory when doing cleanup	Default value = 1000
max_logfile_size	The maximum total size of logfiles kept in the log directory when doing cleanup	Specify <number>[k, kb, m, mb] Default unit = bytes Default value = 1MB

2.5.9 Basic test of the FaMe IIS Plugin

To check whether the FaMe IIS Plugin is installed correctly, edit the *web.config* file to set

```
<add DAD="path2app"
…
version_page="full"
>
```

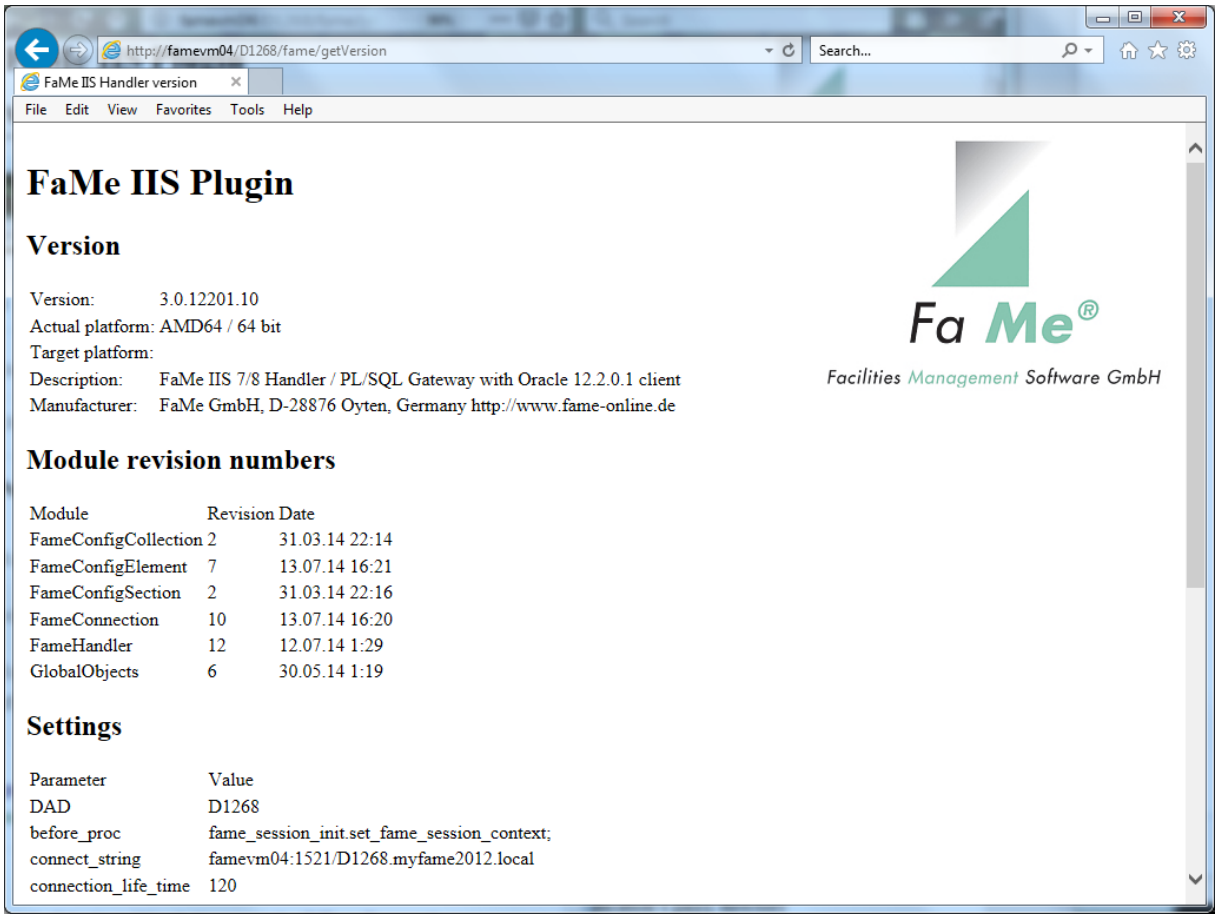
To get the complete version information, where *path2app* is an example for the real URL again.

Point the browser to

```
http://server/path2app/fame/getVersion
```

Again, use your virtual path instead of *server/path2app*. Please note that `getVersion` is case-sensitive.

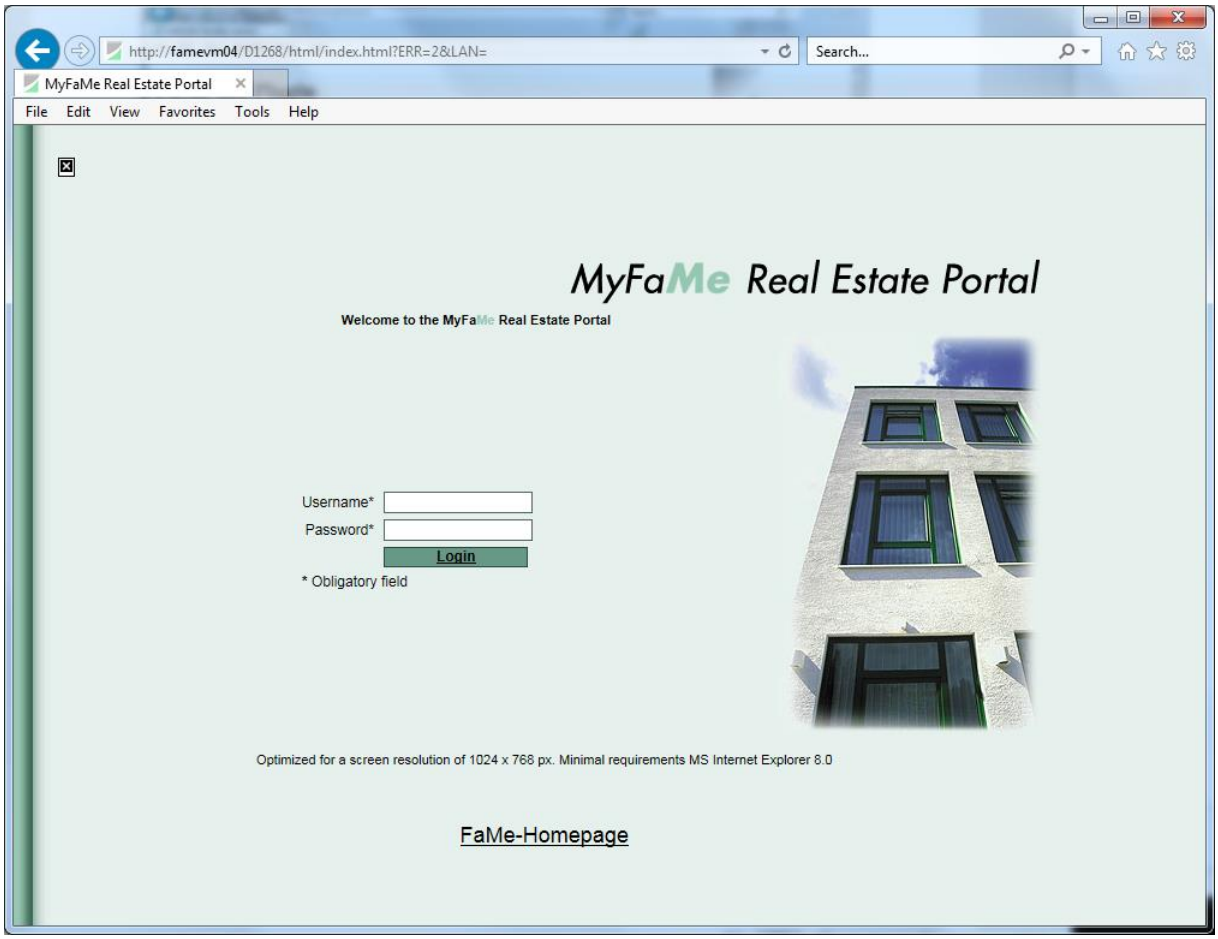
If the FaMe IIS Plugin is working properly and the *version_page* parameter is set to *full* the resulting page will look like this one:



If that works try to call the database directly: either log in, provided you have a FaMe application username and password, or simply call the login procedure without any parameters. *Note*: FAME_WEB is the database schema the FaMe IIS Plugin connects to, not a FaMe application user.

URL:
`http://server/path2app/fame/std_util_pk.login`

Again, replace `/path2app` with your application's virtual path and `server` with the correct server name.



Result of calling the login procedure directly, without any login parameters. If this works, the database responded and the configuration of the FaMe IIS Plugin is complete.

An example for an invalid call to the database:

The FaMe IIS Plugin works, but the URL was wrong.

2.5.10 Troubleshooting: If the FaMe IIS Plugin doesn't work

If there is any error message from the FaMe IIS plugin, check these points:

- Is the *FameHandler.dll* inside the `{document_root}\bin` directory?
  (see *2.5.5 FaMe IIS plugin installation (FameHandler.dll)*

- Is the Oracle client installed? Which version?
  (see *2.4 Oracle client installation*)
- Is the version of *FameHandler.dll* the correct one for the Oracle client installed?
  (see *2.4.2 FaMe IIS Plugin versions and the required Oracle client software version*)
- Is the web.config configured to have
  `<add name="FameHandlerpath2app" path="/path2app/fame" verb="GET,POST" type="FameHandler.FameHandler" resourceType="Unspecified" preCondition="integratedMode" />`
  with path2app replaced by the correct virtual path?
  (see *2.5.8 web.config file*)
- Is there an element
  `<FameConfig> <DADCollection> <add DAD="path2app" …>`
  in the web.config file where "path2app" is replaced by *exactly* the same string that came after *FameHandler*?
- Are the Oracle libraries *oracle.dataaccess.dll* and *oracle.web.dll* linked inside the GAC? (see *2.4.5 ODP.net libraries and their registration in the Windows Global Assembly Cache (GAC)*)

2.6 Tuning the FaMe IIS plugin for performance

There are several parameters for the FaMe IIS Plugin than affect the overall performance of the application. They can be modified in the *web.config* file.
Please note: changing these parameters in the *web.config* file does not require the webserver to be restarted. The changed values will be used for incoming new requests after short time.

2.6.1 Minimum number of database sessions: parameter *min_pool_size*

On sites with higher load it may provide a performance gain to increase the minimum number of database sessions that are permanently available. The default value is 5.

Incrementing this setting will keep additional database sessions open, reducing the database server load as creation of new sessions is a task that requires time and CPU resources on the database server.

2.6.2 Maximum number of database sessions: parameters *max_pool_size* and *connection_timeout*

The *max_pool_size* parameter limits the number of concurrent Oracle database sessions. The default value is 100.

If frequent error messages

Error: Connection to the database failed: ORA-12537: TNS:connection closed

appear in webserver responses the *max_pool_size* parameter and possibly the *connection_timeout* parameter must be increased. The *max_pool_size* parameter is a hard linit for Oracle database sessions held open concurrently. If there are too many parallel requests arriving at the webserver this limit may cause requests to cancelled when the time set by the *connection_timeout* parameter runs out.

2.6.2.1 Increasing the *max_pool_size* parameter

Increasing the *max_pool_size* parameter will require more SGA memory on the database server but allow for a higher number of requests being served in parallel.

2.6.2.2 Increasing the *connection_timeout* parameter

Increasing the *connection_timeout* parameter will only allow for a part of the requests exceeding the limit set by the *max_pool_size* parameter to be processed as this is the time the database connection pool waits for a session to become available.

If *max_pool_size* = 100 is set and there are already 100 sessions busy processing requests any other request will be paused at most *connection_timeout* seconds before being denied with an ORA-12537 error.

Setting this parameter higher than the default value of 15 seconds will usually not improve the server's performance, but it may cause requests to take even longer instead.

On the other hand, setting this parameter to a very low value may cause sessions to be cancelled unnecessarily, causing extra database server workload for frequent creation of new sessions.

2.6.3 The *connection_life_time* parameter

The *connection_life_time* parameter defines the time a session stays open. The default value is 120 seconds.

For earlier versions of the FaMe IIS plugin a setting this low was recommended since lost network connections or server sessions that had crashed or were killed went undetected otherwise. Subsequent requests that tried to use these sessions failed in this case.

The current FaMe IIS Plugin version 3.0.X.18 (with X being the Oracle client version) this is no longer necessary.

2.6.3.1 Increasing the *connection_life_time* parameter

2.6.3.1 Increasing the *connection_life_time* parameter to a longer time will cause the session to be in use longer, reducing the number of new sessions to be created per hour on busy websites (and thus the database server load).

Setting this value to 300 seconds is a safe setting in terms of stability (dead sessions will not be in the pool for more than 5 minutes in this case), but a higher value may increase the overall performance and user experience as creation of new database sessions is a task taking quite some resources on the database server.

2.6.4 Further reading: Oracle documentation on connection pooling

Oracle connection pooling is discussed in detail on this page:

https://docs.oracle.com/en/database/oracle/oracle-database/19/odpnt/featConnecting.html#GUID-AAC5352A-83F2-483B-A681-93A5200CA83A

2.7 Debug log

The FaMe IIS plugin has a debug option for diagnosing and testing purposes. To activate it use the *fame_debug_mode* parameter.

2.7.1 The *fame_debug_mode* parameter

The *fame_debug_mode* parameter can be set to one of these values:

***fame_debug_mode* setting**	**Effect**
off	No debugging, but crash logfiles get written
On	A summary is written for each request, including all parameters
user	Each request is logged in detail, including the response (except for downloads from FaMe document management)
session	Like user, but only active of the user activates the debug mode for his or her session manually – see below. Introduced with FaMe IIS Plugin 3.0.X.14.

2.7.2 The *fame_debug_path* parameter

The *fame_debug_path* parameter sets the directory for the logfiles.

If the path is invalid or is not writable by the *Application pool identity*, no logfiles will be generated, and there will be no notice about that.

2.7.3 Disabling the generation of logfiles altogether

Even with the *fame_debug_mode = off* setting small logfiles will be generated when the FaMe IIS Plugin encounters an application error that is not handled. Those files have a file ending of *.crahs.log*.

If this behaviour is not desired, set the *fame_debug_path* parameter to an empty or invalid path, write protect the directory, or disallow the *Application pool identity* to create files there.

2.7.4 Activating the debug per session

On busy websites activating the debug log in general by setting *fame_debug_mode = on* or *fame_debug_mode = user* is discouraged as it may create a large number of files, slowing down the processing of the requests as the number of files grows.

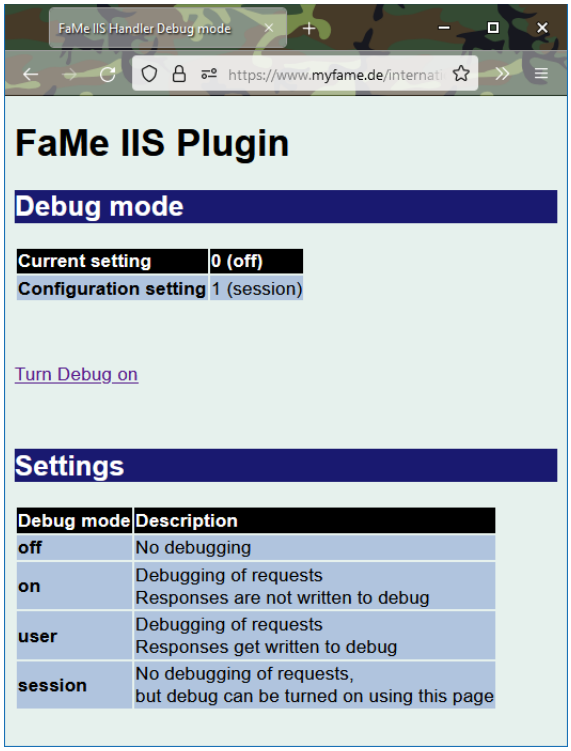
The parameter *fame_debug_mode* now has a new option *session*. This was introduced with version 3.0.X.14. This enables the debug mode to be enabled for each session individually.

If *fame_debug_mode=session* is set in the web.config file users can enable the debug for their session from the browser by opening the following page

`http://server/path2app/fame/setDebug`

The filename `setDebug` is case-sensitive.

Clicking on the link will turn debugging on resp. off.
Accessing the files will require file access to the web server.



2.7.5 Automatic logfile cleanup

The FaMe IIS Plugin will clean up old logfiles by several criteria:

- By age
- By total size
- By number

The cleanup is done when a new request is processed. Whether to do a cleanup run is determined by the timestamp of a file named *cleanup* in the log directory. If this file is older than the setting of the *cleanup_log_interval* parameter's setting and no other process has this file open files in the log directory are evaluated for deletion.

After deleting old logfiles a short summary is written into the *cleanup* file.

The cleanup operation is stopped after 5 seconds in cases where a high number of logfiles has accumulated.

To automatic cleanup is not in use if the parameter *fame_debug_mode* is set to *off*.

Parameters controlling the automatic cleanup:

max_logfile_age	The maximum age of logfiles after which a file is to be deleted by cleanup operation	Specify <number>[dhm] to specify a max. age of <n>days, hours, or minutes. Default unit = minutes Default value = 1d
cleanup_log_interval	The interval after which a new logfile cleanup is started.	Specify <number>[dhm] If the *cleanup* file inside the log directory is missing or older than this value old logfiles in this directory are deleted. Default unit = minutes Default value = 1d
max_logfile_count	The maximum number of logfiles kept in the log directory when doing cleanup	Default value = 1000
max_logfile_size	The maximum total size of logfiles kept in the log directory when doing cleanup	Specify <number>[k, kb, m, mb] Default unit = bytes Default value = 1MB

2.8 Setting up the FaMe IIS Plugin for Windows Single-Sign-on (SSO) configuration

The FaMe application supports Windows Single-Sign-on (SSO), allowing for Windows domain users logged in at the client workstation to use the FaMe application without having to log in with a separate username and password which would be managed internally by the database application.

Instead, the users open the login page without specifying the username and password, and the IIS server recognizes the Windows account used on the client by its current session. The username and domain name are forwarded to the login procedure which looks up both parameters in the FaMe user table. If there is a FaMe user account identical to the Windows account name, and it does also have the same domain name plus the *External identification* flag set, the login is accepted.

Advantages of Windows SSO over FaMe-managed accounts:

- No password needs to be kept in the database
- The user's Windows username is used by FaMe as well
- Password validation and account expiration are managed by the Windows domain

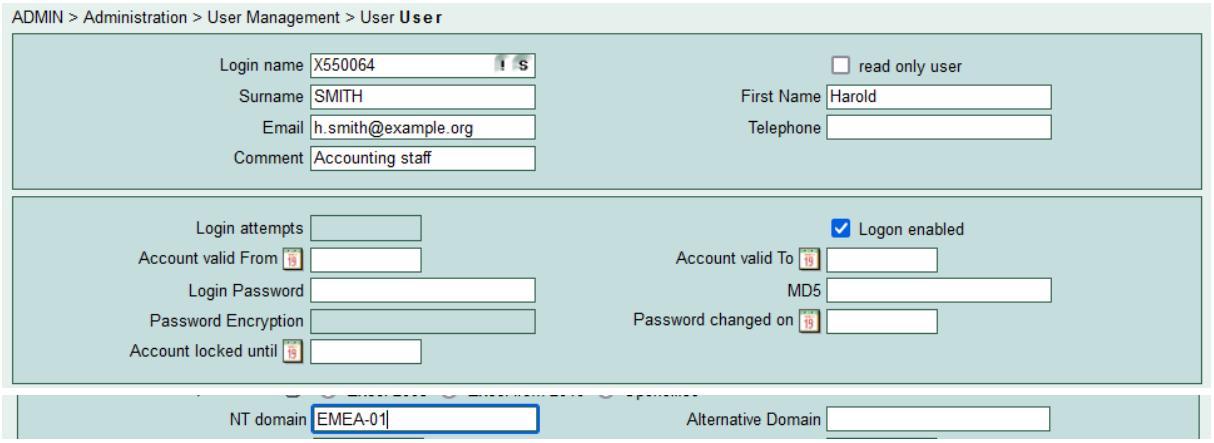
2.8.1 Setup of FaMe user accounts

FaMe user accounts are managed by the FaMe application independently of the use of the SSO mechanism. Using SSO removes the need for passwords managed and verified internally by the FaMe application.

It requires the FaMe user account names to be identical with the Windows account names.

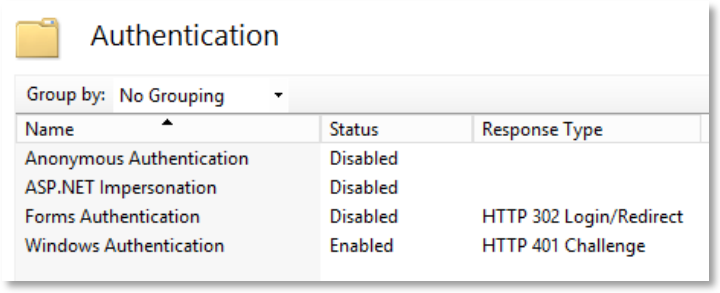
To enable FaMe users to log in with their Windows accounts, these settings must be made in the FaMe user management for each user who will use SSO:

Parameter	Setting	Description
Login name	Equal to Windows domain username	The Windows account name to use
Logon enabled	Set	Accounts may be disabled to log in to FaMe even if the Windows account is valid in the Windows domain
NT domain	Equal to Windows domain	Is matched at login. A second domain may be specified as *Alternative domain*
Alternative domain	Equal to another Windows domain, optional	Used to specify another domain if users from 2 Windows domains use the application with SSO login



2.8.2 IIS configuration for SSO

On the IIS side it is important to enforce *Windows authentication* in the IIS configuration.



This setting must be done for the virtual path `/application/fame`. It may be done for `/application` or the entire site (path `/`) if this setting is inherited by the virtual path `/application/fame`.

2.8.3 Enabling login for Windows users outside the domain or a with FaMe-managed account when Windows SSO is in use

If Windows authentication is configured for the virtual path `/application/fame`, it is required for all users to be logged in to the IIS server's Windows domain on the client computer or to log in to that domain when prompted in the browser.

2.8.3.1 Case 1: Application users that have a valid Windows domain account but are not logged in to Windows currently

If the user at the client computer is not logged in to the domain at Windows level, the IIS will respond with a HTTP-401 Unauthorized response. The browser now prompts for Windows login:

This dialog is shown by the browser when the IIS did not recognize the Windows session used at the client, for instance, when the client is logged in to a local Windows account (or using a non-Windows device).

With valid Windows credentials the SSO login can be completed now.

2.8.3.2 Case 2: Application users without a Windows domain account

For users not logged in to the Window domain a separate virtual path without Windows authentication and hence without SSO must be configured. To do so, the application's files are either copied or aliases are set for all the application's directories.

If the application uses the virtual path `/application/fame`, it may be desirable to have another virtual path `/application2/fame` which does *not* use Windows authentication. Copying the entire application directory (typically `c:\inetpub\wwwroot\application` in this case) to a new directory, say, `c:\inetpub\wwwroot\application2`, and turning off Windows authentication for this directory will suffice to implement this.

Users logging in using this alternate path will need FaMe-managed passwords. Those passwords are kept as *salted hash* values in the FaMe user table and verified by the login procedure. To set these passwords the FaMe application's user administration page must be used.

2.8.4 SSO support for multiple domains

SSO with more than one Windows domain will require one of two options:

- Setting up a second IIS server inside any of the other domains
- Setting up a Windows domain trust relationship between the primary domain (the one the webserver is a member of) and any other domain that shall be supported.
  This second case is currently not covered in this document.

The first option allows the domains to be kept separate but requires for multiple IIS servers to be managed.

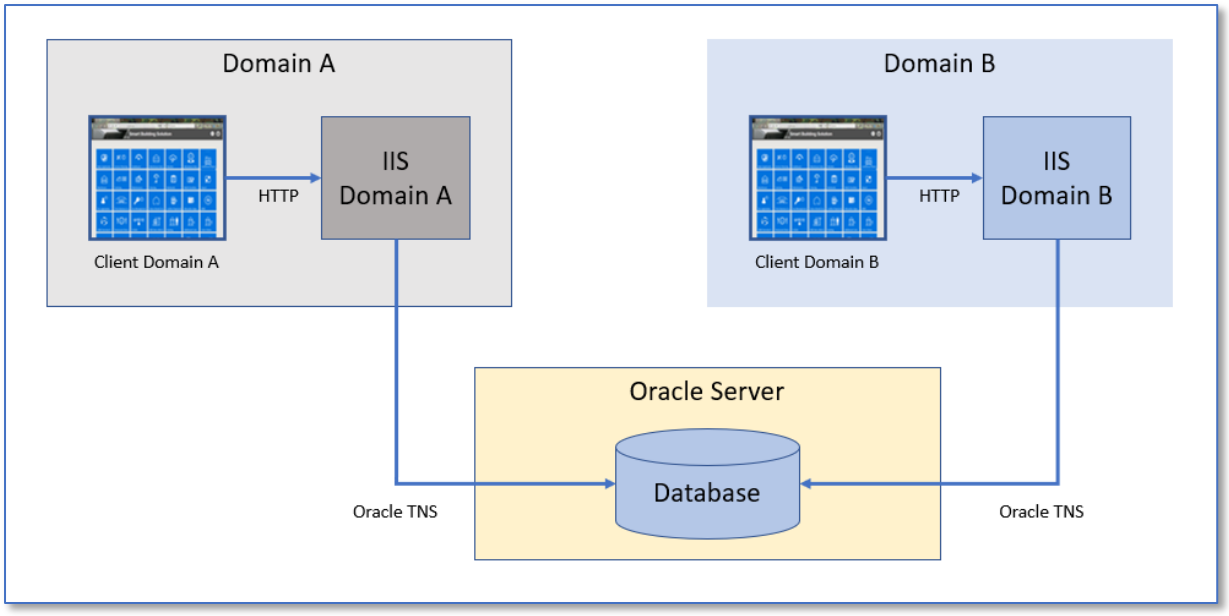
The second option requires extra permission management to keep the domains apart as the trust relationship between 2 domains opens up both domains for all users.

If both options are considered to be too much administrative effort it is recommend to either

- provide users from domain B with accounts in domain A (domain A contains the IIS server here)
- or let users outside of domain A log in with FaMe-managed passwords (this requires a second virtual path on the IIS server that does *not* use Windows authentication; see above)

2.8.5 Setting up additional webservers to support multiple-domain SSO

The setup with 2 webservers in 2 different Windows domains is shown here:



Each domain requiring Windows authentication is running a web server of its own to avoid setting up a trust relationship in this case.

Clients from any domain are using their own web server, allowing for Windows SSO.

The setup for both webservers is identical except for their domain membership.

The database server is basically independent of any domain as the webservers connect to Oracle using a local Oracle client and the Oracle TNS protocol.

Note: on the database server it is required to change the SQLNET.AUTHENTICATION_SERVICES setting in the sqlnet.ora file to

```
SQLNET.AUTHENTICATION_SERVICES=(none)
```

With the original SQLNET.AUTHENTICATION_SERVICES=(NTS) setting for this parameter the Oracle server will require the client (the webserver in this case) to be in the same Windows domain as the server.

As a side effect, this setting will disable OS authentication when managing the Oracle database. The original NTS setting is required when database administrators want to log using `sqlplus "/ as sysdba"`

Setting SQLNET.AUTHENTICATION_SERVICES=(NONE) will require DBAs to specify the SYS password.

2.9 PDF Writer and Graphics module installation

The FaMe PDF Writer is a Perl CGI program that formats FaMe report data as PDF documents. It uses HTML template files that define the formatting of a report and HTTP access to a PL/SQL package inside the database.

2.9.1 Requirements

On the web server the following components are required:

- Script `pdf_report.pl`, to be installed in the application's `perl` directory
- The FaMe Perl library
- A Perl Interpreter and runtime library, e. g. ActiveState Perl, Strawberry Perl
- The *PDFlib* library for Perl; for the production system a license key is required as well
- The application's HTML templates
- A directory writable by IIS user processes for temporary files and logfiles
- Anonymous HTTP access to the FaMe IIS Plugin
- If the Graphics module is to be installed, it will require access to the Oracle client
- The *Perl Chartdirector* library if the Graphics module is to be installed

2.9.2 Installation package

For FaMe PDF Writer version 2.4 the latest version is `FaMe_FM5_PDF_Writer_2.4.2.2.zip`.

2.9.3 Additional documentation

About the FaMe PDF Writer the following additional documents are available in German only:

- FaMe-FM5.pdf_report_generator.dokumentation.2.3.pdf
- FaMe-FM5.pdf_report_generator.reference.pdf
- FaMe-FM5.pdf_report_generator.troubleshooting.txt

The troubleshooting guide is available in English as well:

- FaMe-FM5.pdf_report_generator.troubleshooting.en.Rev.0.txt
- FaMe-PDF-Writer-diagram-of-installation.ppt

For documentation check the `doc` folder of the installation package.

2.9.4 Database access by the FaMe PDF Writer

The PDF Writer sends HTTP requests to its own web server to access the database, requiring the database to have the PDF_SUPPORT_PAK package installed which is the PDF Writer's database interface.

The database interface is, like all database procedures, mapped to the virtual path `fame`.

This virtual path cannot use Windows authentication, so anonymous access is required.

2.9.4.1 Special case of Integrated Windows Authentication for the virtual path *fame*

If a Windows SSO configuration and hence Integrated Windows Authentication is in use for

this virtual path a 2nd virtual path must be set up using the FaMe IIS Plugin using anonymous access.

Example:

If the database procedures reside in `/path2app/fame` and this virtual path uses Integrated Windows authentication, another virtual path, e.g. `/path2app/famepdf` using the FameHandler may be configured. The configuration is identical to the configuration of `/path2app/fame`, but it uses anonymous access. In the `pdf_report.conf` file (see below) the parameter `path2db` must be set to use this 2nd virtual path:

```
path2db=../famepdf
path2doc=../famepdf/doc
```

2.9.5 Migration to newer Windows Server versions and PDFlib licensing

If the ongoing installation is a migration to a newer Windows Server version it may be desirable to preserve an existing PDFlib license. If the installed PDFlib version is 8.X the newest Perl version supported is 5.12. But as Perl 5.12 may not run properly on Windows Server 2012 an upgrade to PDFlib 9 is recommended.

2.9.6 Perl interpreter and runtime environment

Basically, any newer Perl interpreter suitable for Windows 2012 will be fine, but the runtime environment must contain a number of Perl (CPAN) modules which have to be installed if the runtime environment used does not provide them out of the box.

The Perl version should be **newer than 5.12** as some problems with file access have been observed with Perl 5.12 on Windows Server 2012.

This manual describes how to use a current ActiveState Perl distribution which contains most Perl modules required by the FaMe PDF Writer.

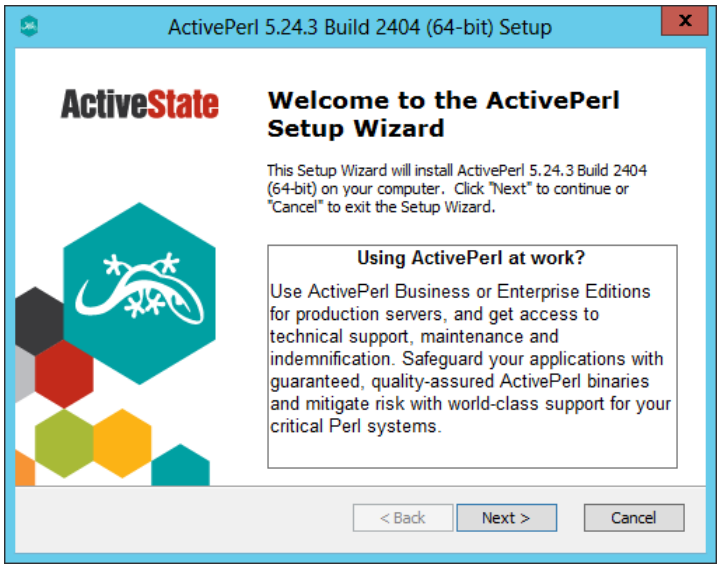
ActiveState Perl may be downloaded from

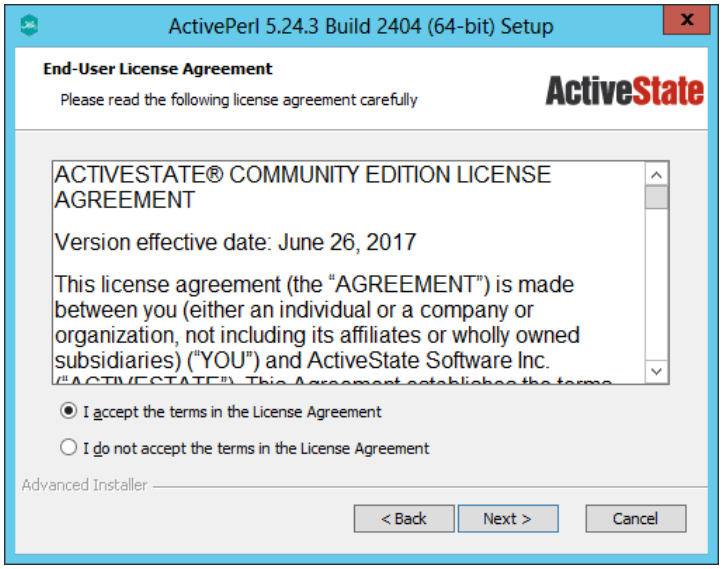
https://www.activestate.com/activeperl

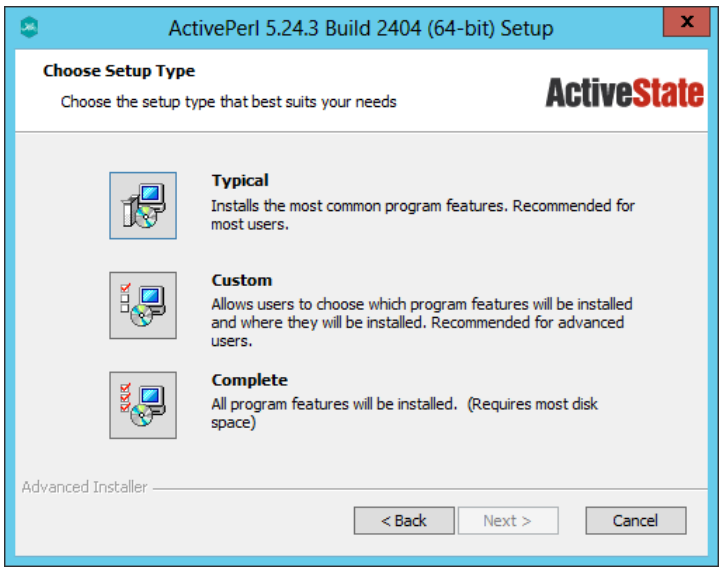
The Perl version used in the installation described here is 5.24. Which version should actually be used also depends on the *PDFlib* version as the PDFlib installation package comes with precompiled Perl modules that must match the Perl interpreter's version to the first 2 levels: if Perl 5.24 is installed the PDFlib library for Perl 5.24 must be installed.

Installation process:

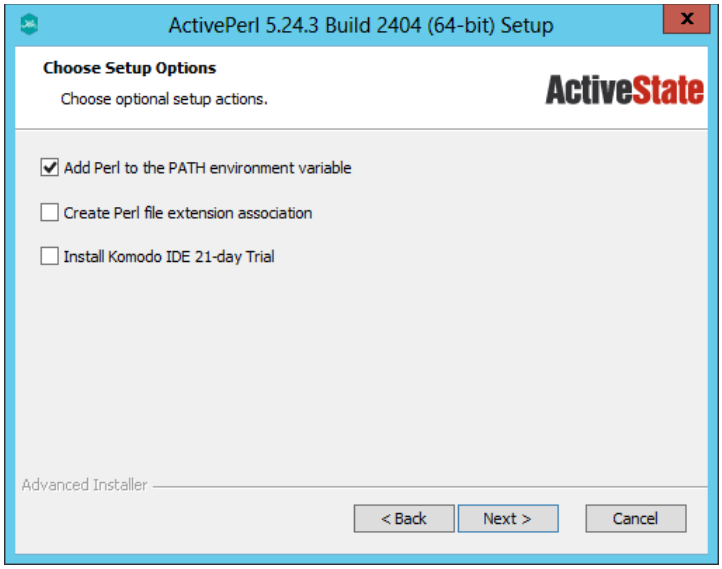
The *typical* installation option as shown here will install Perl in the directory `c:\Perl64`.

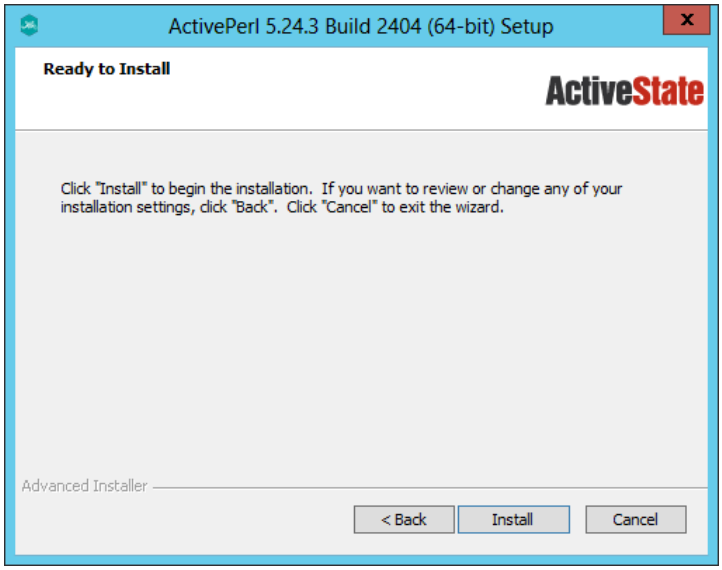


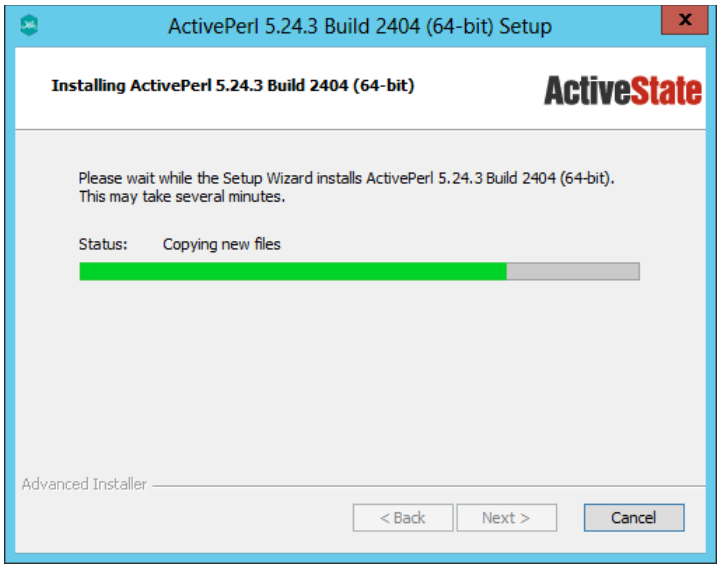


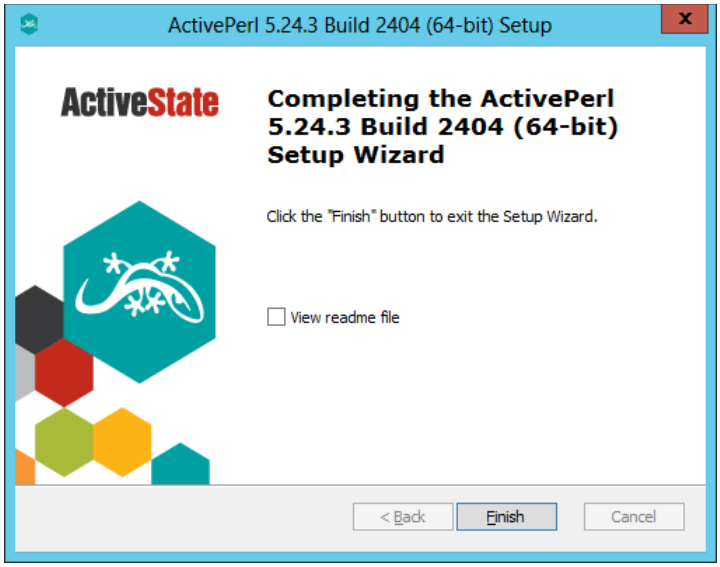


Typical was the option selected in the installation documented here.


ActivePerl 5.24.3 Build 2404 (64-bit) Setup
Choose Setup Options
Choose optional setup actions.
ActiveState
Add Perl to the PATH environment variable
Create Perl file extension association
Install Komodo IDE 21-day Trial
Advanced Installer
< Back
Next >
Cancel


ActivePerl 5.24.3 Build 2404 (64-bit) Setup
Ready to Install
ActiveState
Click "Install" to begin the installation. If you want to review or change any of your installation settings, click "Back". Click "Cancel" to exit the wizard.
Advanced Installer
< Back
Install
Cancel

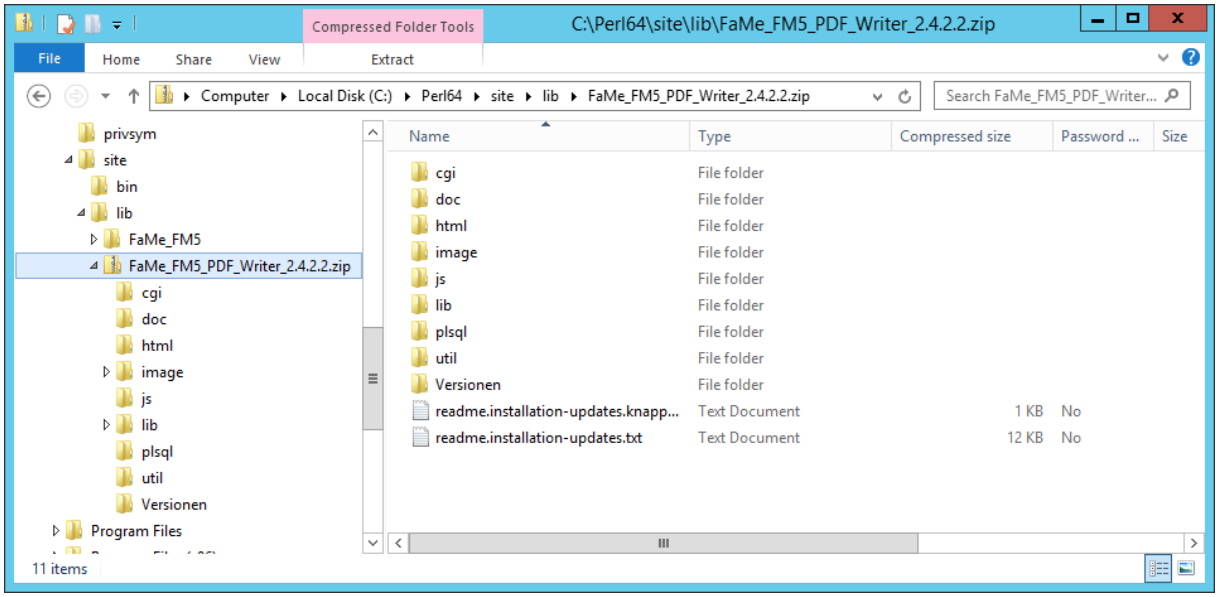

ActivePerl 5.24.3 Build 2404 (64-bit) Setup
Installing ActivePerl 5.24.3 Build 2404 (64-bit)
ActiveState
Please wait while the Setup Wizard installs ActivePerl 5.24.3 Build 2404 (64-bit). This may take several minutes.
Status: Copying new files
Advanced Installer
< Back
Next >
Cancel



2.9.7 FaMe Perl library and script `pdf_report.pl`

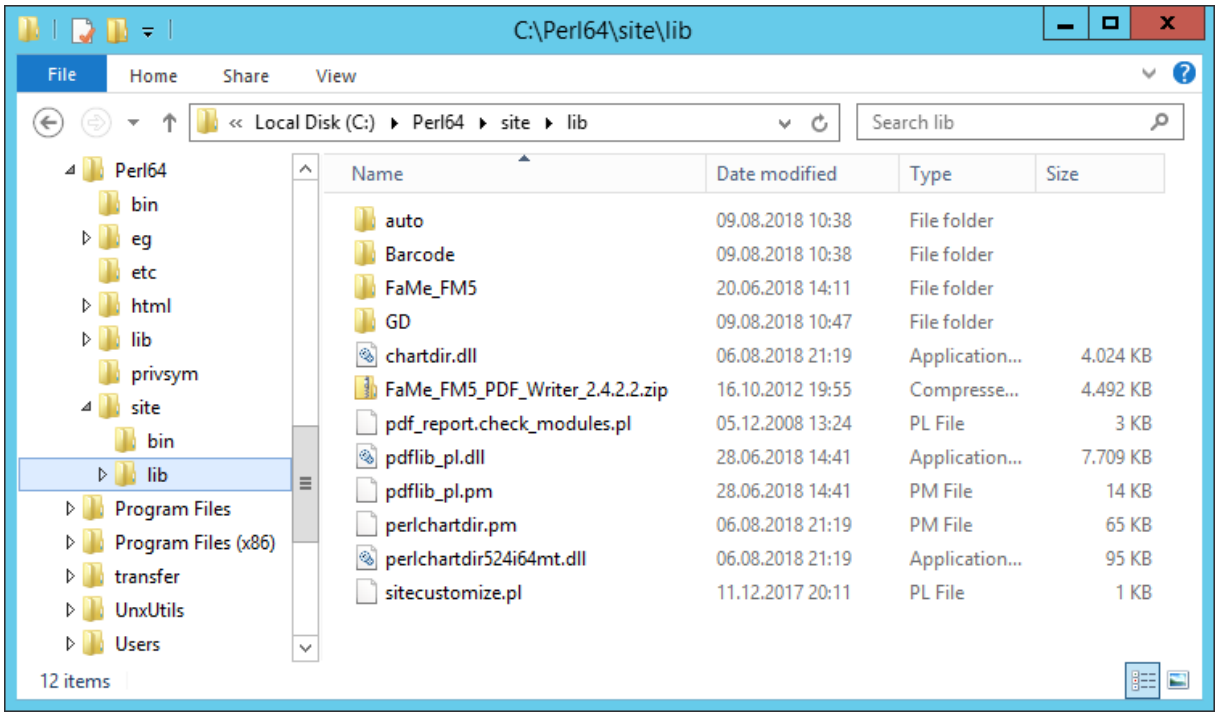
The FaMe Perl library is provided as a zip file and is to be unzipped into an arbitrary directory. While basically any directory may be used, the `site\lib` directory within the Perl runtime environment's installation directory is preferred as this eliminates the requirement to set the path to additional Perl libraries in the IIS configuration for Perl. Using the `site\lib` directory eliminates the requirement to set Perl library paths in the IIS configuration.

For PDF Writer version 2.4 the installation package is `FaMe_FM5_PDF_Writer_2.4.2.2.zip:`



2.9.7.1 FaMe Perl Library

The installation package contains the directory `lib\FaMe-FM5`. This directory must be copied into the `site\lib` directory:



This screenshot shows the complete set of libraries used by the FaMe PDF Writer and FaMe Graphics Module.

2.9.7.2 Verifying the installation of the FaMe Perl library

The installation of the FaMe Perl library can be verified using the script `pdf_report.check_modules.pl` which is part of the installation package. It must be run by the Perl interpreter from the command line:

```
C:\Perl64\site\lib>perl pdf_report.check_modules.pl
Archive::Zip    OK
AutoLoader      OK
Barcode::Code128        OK
CGI     OK
CGI::Cookie     OK
CGI::Util       OK
Carp    OK
Compress::Zlib  OK
Config  OK
Cwd     OK
Data::Dumper    OK
Digest::MD5     OK
Digest::base    OK
DynaLoader      OK
Encode  OK
Encode::Alias   OK
Encode::Config  OK
Encode::Encoding        OK
Errno   OK
Exporter        OK
Exporter::Heavy OK
FaMe_FM5::pdf_report::config    OK
FaMe_FM5::pdf_report::form_tree OK
FaMe_FM5::pdf_report::mail      NOT PRESENT!
FaMe_FM5::pdf_report::pdf_data  OK
FaMe_FM5::pdf_report::pdf_writer        OK
FaMe_FM5::pdf_report::pdf_writer::barcode       OK
FaMe_FM5::pdf_report::pdf_writer::functions     OK
```

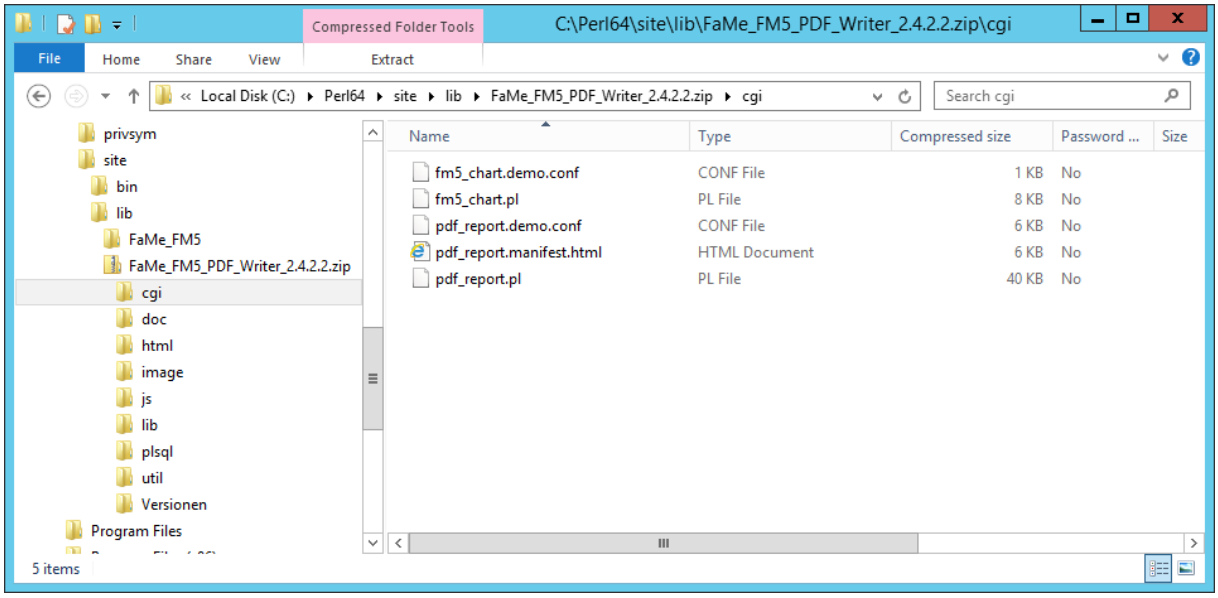
```
FaMe_FM5::pdf_report::pdf_writer::print_params  OK
FaMe_FM5::pdf_report::pdf_writer::stack OK
FaMe_FM5::pdf_report::report_config     OK
FaMe_FM5::pdf_report::template  OK
FaMe_FM5::pdf_report::template_element  OK
FaMe_FM5::util::csvdata OK
FaMe_FM5::util::messages        OK
FaMe_FM5::util::standard        OK
FaMe_FM5::util::unicode2entity  OK
FaMe_FM5::vendor        OK
Fcntl   OK
File::Basename  OK
File::Copy      OK
File::Find      OK
File::Path      OK
File::Spec      OK
File::Spec::Unix        OK
File::Temp      OK
FileHandle      OK
GD      OK
GD::Barcode     OK
HTML::Entities  OK
HTML::Parser    OK
HTML::PullParser        OK
HTML::Tagset    OK
HTML::TokeParser        OK
HTTP::Date      OK
HTTP::Headers   OK
HTTP::Message   OK
HTTP::Request   OK
HTTP::Request::Common   OK
HTTP::Response  OK
HTTP::Status    OK
Hash::Util      OK
IO      OK
IO::File        OK
IO::Handle      OK
IO::Seekable    OK
LWP     OK
LWP::Debug      OK
LWP::MemberMixin        OK
LWP::Protocol   OK
LWP::UserAgent  OK
List::Util      OK
MIME::Base64    OK
MIME::QuotedPrint       OK
Mail::Sender    NOT PRESENT!
POSIX   OK
Scalar::Util    OK
SelectSaver     OK
SelfLoader      OK
Socket  OK
Symbol  OK
Time::HiRes     OK
Time::Local     OK
URI     OK
URI::Escape     OK
XML::Simple     OK
XSLoader        OK
base    OK
bytes   OK
constant        OK
integer OK
```

```
open     OK
overload         OK
pdflib_pl        OK
re       OK
strict   OK
utf8     OK
vars     OK
warnings         OK
warnings::register       OK
One or more FaMe_FM modules not found.
Use perl -I<path> to include the FaMe module library.
```

This test shows that most of the Perl modules are installed, including the FaMe Perl library (modules named *FaMe_FM5::...*) and the PDFlib library. Some modules are optional and not installed, like the mail modules.

2.9.7.3 Script pdf_report.pl

The package contains the script `pdf_report.pl` as well. It's version must be the same as in the installation package, so it must be copied from the zip archive's `cgi` folder into the application's `perl` folder:



Copy `pdf_report.pl` *to* `c:\inetpub\path2app\perl`*, provided c:\inetpub\path2app is the application folder of your application.*

If the version of the PDF writer has been modified, it may be necessary to upgrade the configuration file, too. To do this, run the script pdf_report.pl from the command line specifying the parameter `-upgrade-config`:

```
C:\inetpub\wwwroot\D1047\perl>c:\Perl64\bin\perl pdf_report.pl --upgrade-
config
FaMe-FM5 PDF Writer Version 2.4.2.2, Date 2012 10 16
Warning: file ./pdf_report.conf exists!
OK to update/overwrite? (Y/N)  Y
Reading file ./pdf_report.conf ....
Invalid section, eliminated:
[Page layout]2 new parameters were added to this configuration

Configuration file ./pdf_report.conf has been updated.
3 Changes.
```

```
C:\inetpub\wwwroot\D1047\perl>
```

2.9.7.4 Static web files for the FaMe PDF Writer

The installation package contains 3 subfolders `html`, `image`, and `js` with HTML, Javascript, and images for the PDF writer. Their contents must be copied into the corresponding application directories:

```
FaMe_FM5_PDF_Writer_2.4.2.2.zip\js        => c:\inetpub\wwwroot\path2app\js
FaMe_FM5_PDF_Writer_2.4.2.2.zip\image     =>
c:\inetpub\wwwroot\path2app\image
FaMe_FM5_PDF_Writer_2.4.2.2.zip\html      =>
c:\inetpub\wwwroot\path2app\html
```

2.9.8 PDFlib library

The *PDFlib* library is used by the FaMe PDF Writer to create PDF files from preformatted data. It is a commercial library created and distributed by

PDFlib GmbH
Franziska-Bilek-Weg 9
D-80339 München
Germany
http://www.pdflib.com/

The PDFlib library must be licensed for each server it is installed on. At the time this document's version was created the licensing model stated that test systems may use a license without extra costs. Running PDFlib without a license key will result in watermarks printed on each page produced.

2.9.8.1 Download of the PDFlib installation package

Download of the PDFlib library is possible from PDFlib's website:

http://www.pdflib.com/download/pdflib-product-family/

Choose
Windows Server x64 and Windows XP/Vista/7/8/10 x64
from this line of packages choose *Perl*.

As of this documentation, the current Installation package is
`https://www.pdflib.com/binaries/PDFlib/912/PDFlib-9.1.2p1-MSWin64-perl.zip`

2.9.8.2 Installation

The installation of the PDFlib library is a manual process. There are 2 files that must be copied from the PDFlib installation package into the `site\lib` folder of the Perl environment. This is the same folder into which the FaMe Perl library has been copied before.

Files to copy:

- bind\perl\pdflib_pl.pm
- bind\perl\[distribution]\pdflib_pl.dll

The correct version of pdflib_pl.dll depends on the Perl version and distribution installed. For ActiveState Perl 5.24 the correct dll version is inside the installation package under bind\perl\ASPerl5.24.

The screenshot shown in paragraph *2.9.7.1 FaMe Perl Library* shows the location where these

files must be copied into.

2.9.8.3 PDFlib license file

The license key must be readable by the PDFlib library at runtime, otherwise it will print a diagonal watermark over each page produced saying “www.pdflib.com”.

The license key is to be copied into a text file supplied with the installation package where it is named `licensekeys.txt`.

Contents of this file:

```
PDFlib license file 1.0

# This is a license file template for PDFlib GmbH products.
# Replace the 0 in the third column with your actual license key.

PDFlib      9.1.2          0
```

The actual path and filename of this file must be specified in the file `pdf_report.conf`. See below.

2.9.9 Barcode libraries GD::Barcode and Barcode::Code128

To print barcodes the FaMe PDF Writer uses the Perl modules GD::Barcode and Barcode::Code128. These are not included with the *ActiveState Perl* distribution and have to be added manually. They are provided by FaMe as a separate package named *perl-barcode.zip*.

To install these modules, copy the 2 folders `GD` and `Barcode` into the `site\lib` directory of the Perl environment. This is the same folder into which the FaMe Perl library has been copied before.

2.9.10 Script pdf_report.pl and configuration file pdf_report.conf

The script `pdf_report.pl` is the CGI frontend of the FaMe PDF Writer and must be copied into the `perl` folder of the application (`c:\inetpub\wwwroot\<appdir>\perl`) with `<appdir>` being the application folder.

The configuration file `pdf_report.conf` must be copied into the same directory. It is a plaintext file that defines parameters for the script pdf_report.pl.

The screenshot shown in paragraph *2.9.7.1 FaMe Perl Library* shows the location where these files must be copied into.

2.9.10.1 Script start_pdf_report.pl

In some cases, the Perl interpreter may print warnings on STDERR output which will cause the IIS to stop the script and return a HTTP-502 error to the client.

To avoid that, an extra script named `start_pdf_report.pl` is supplied with the installation package. This script starts another process running the Perl interpreter and the script pdf_report.pl, sending all STDERR output to NUL:

```
if (system("C:\\Perl64\\bin\\perl.exe s_pdf_report.pl 2>NUL") != 0) {
   print "content-type: text/plain\n\n";
   print STDERR "s_pdf_report.pl: $err\n";
   print <<EOF
Status:  500 error running s_pdf_report.pl
content-type: text/plain
…
 }
```

This script is optional and should only be used if the PDF Writer either permanently or sporadically fails with HTTP-502 errors. The original script `pdf_report.pl` shall be renamed to `s_pdf_report.pl` in this case, and `start_pdf_report.pl` becomes `pdf_report.pl`

- `pdf_report.pl => s_pdf_report.pl`
- `start_pdf_report.pl => pdf_report.pl`

2.9.11 Basic PDF Writer configuration

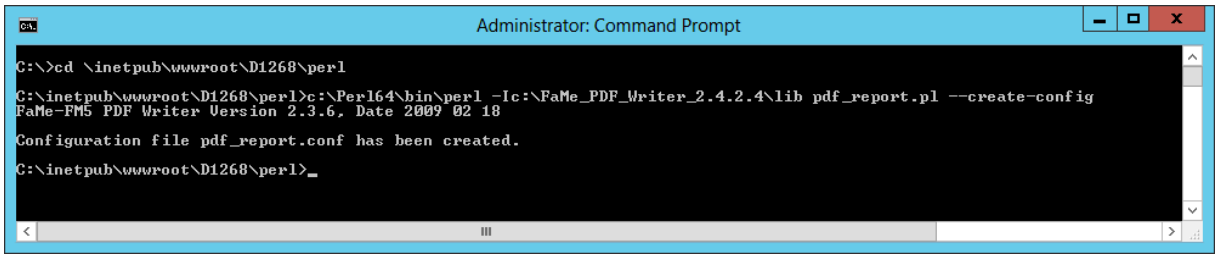
The configuration of the PDF Writer is kept in the file *pdf_report.conf*.

This file is a plain text file and must be copied into the *perl* folder of the FaMe application directory.

An initial *pdf_report.conf* file can be created with this command line:

```
c:\Perl64\bin\perl -Ic:\FaMe_PDF_Writer_2.4.2.4\lib pdf_report.pl --create-config
```

Please note that here the path to the FaMe PDWF Writer's library must be set correctly



2.9.11.1 Minimal pdf_report.conf file

Comment lines in this file start with a pound sign.

```
# pdf_report.conf example file with minimal settings
cleanup_interval = 1d
disk_template_directory = ../html/tmpl
font_path = c:\windows\Fonts
http_charset = UTF-8
image_directory = ../image
language = english
log_directory = logs
log_file = pdf_report_$REMOTE_ADDR.$DATETIME-ISO8601.log
log_mode = debug
path2db = ../fame
path2docs = ../fame/doc/
pdf_charset = UTF-8
pdf_filename_schema = $REPORT_$DATETIME-ISO8601_$$.pdf
pdflib_license_file = c:\fame\pdflib\pdflib.lic
session_cookie_name = SID
system_charset = UTF-8
template_file_schema = tmpl_%s.html
template_form_header = multilanguage/tmpl_form_header.html
zip_logs = 1
```

2.9.11.2 Basic parameters in pdf_report.conf

The basic parameters are described here, and the important ones that must be checked are marked *YES* in the *Check* column:

Parameter	*Description*	*Check*	*Standard value*
cleanup_interval	The time after which old logfiles will be deleted by the next run of the PDF writer		"1d" (1 day)
default_page_layout	Default page size specify DIN page sizes here; for others see the PDF Writer's main manual		A4
disk_template_directory	The local directory where HTML report templates are installed	YES	../html/tmpl
font_path	The path to the local installation of TTF fonts the PDF writer will use On Windows, the usual setting is C:\Windows\Fonts	YES	C:\Windows\fonts
http_charset	Character set used for HTTP access to the database. If the database uses Unicode, this parameter should be set to UTF-8	YES	UTF-8
image_directory	Local directory for additional image files that appear in PDFs	YES	../image
log_directory	Log directory Must be writable by server processes run under IIS. Used for logs as well as temporary files. **This directory need read and write permissions for all Windows users the PDF Writer runs as. See 2.9.13.3** Conversion of the perl folder to an application**.** **See next paragraph.**	YES	logs
log_file	Name schema for logfiles May contain placeholders for date, time, IP address, process ID		pdf_report_ $LOG_FILE_NUM.log
log_mode	none/log/debug Log mode used by the PDF Writer Use 'debug' for troubleshooting and 'none' for production		'none'
path2db	Virtual path to the database's PL/SQL-Gateway or the FaMe IIS Plugin. Should always be a relative path.	YES	../fame
path2docs	Virtual path to the database's document storage; analogous to *path2db*	YES	../doc
pdf_charset	Character set for PDF documents created If the database uses Unicode, this parameter should be set to UTF-8	YES	UTF-8
pdf_filename_schema	Schema for filenames used for the PDFs created May contain placeholders for date, time, user		$REPORT_$DATETIME-ISO8601_$$.pdf

pdflib_license_file	The location and filename of the file containing the PDFlib license key		
system_charset	Operating system's character set **Windows:** usually Codepage 1252 (cp1252) Western Europe/USA. Preferrably set to UTF-8 as well on Windows to avoid conversion warnings. **Linux/UNIX:** UTF-8 in most cases	YES	UTF-8
template_file_schema	Schema for filenames derived from report names. Used to find template files for report formatting inside the directory defined by *disk_template_directory* (See above)		tmpl_%s.html
template_form_header	Name of the HTML template file containing the default page header, footer, and 1st and last page.		tmpl_form_header.html
zip_logs	1 = zip logfiles Requires optional Perl module Archive::Zip which actually comes with ActiveState Perl. Silently ignored if Archive::Zip is not installed.		1

2.9.12 The log directory for the FaMe PDF Writer

The FaMe PDF Writer uses a directory for logfiles and as well temporary files. This directory must provide read, write, and delete permissions for every user under which the PDF writer is run. Delete permission is required as temporary files are cleaned up after use, and logfiles after an interval configured by the *cleanup_interval* parameter in the *pdf_report.conf* file.

If anonymous access to the *perl* directory is configured in IIS it is the application pool user that requires write permission. Provide full control permissions for that user on this folder.

The log directory is the one configured by the *log_directory* parameter in the *pdf_report.conf* file.

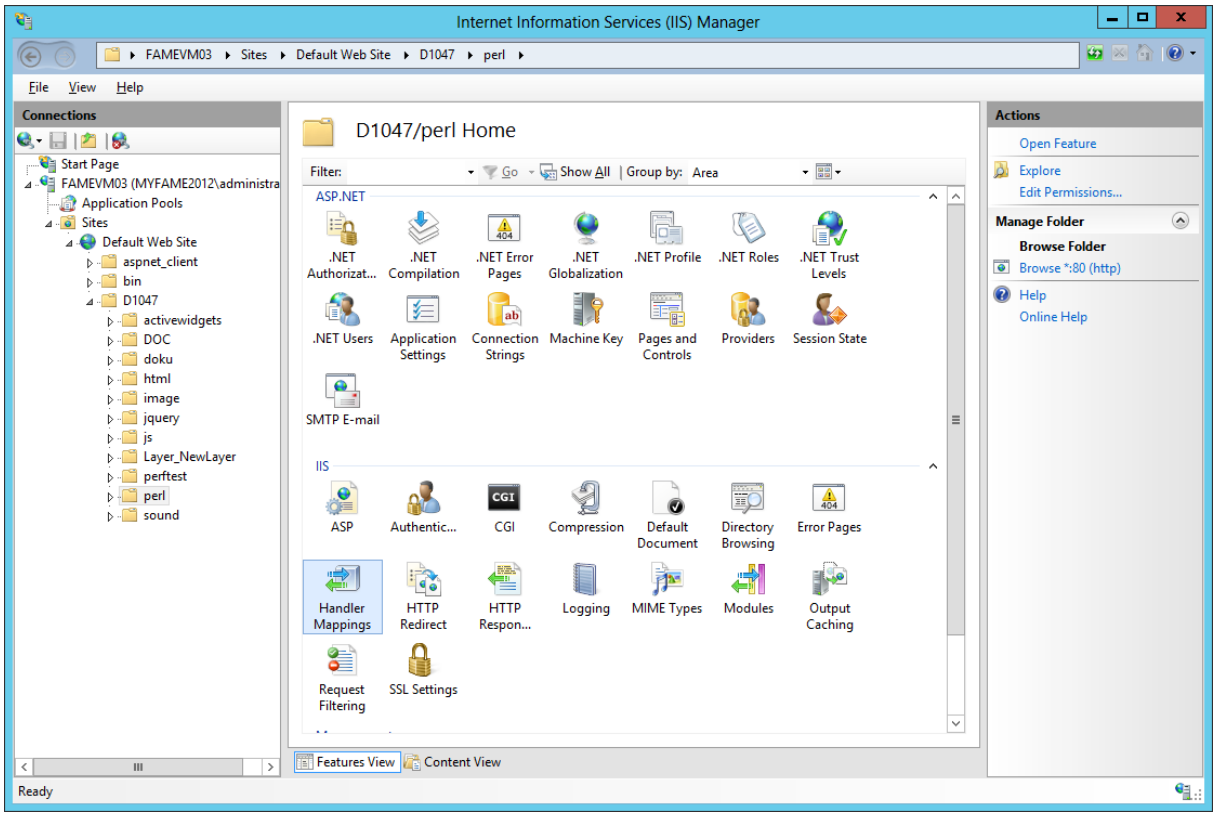
2.9.13 IIS configuration for Perl scripts

The IIS must be configured to run Perl scripts in the application's *perl* folder. The configuration can be done using the IIS manager or by editing the *web.config* file.

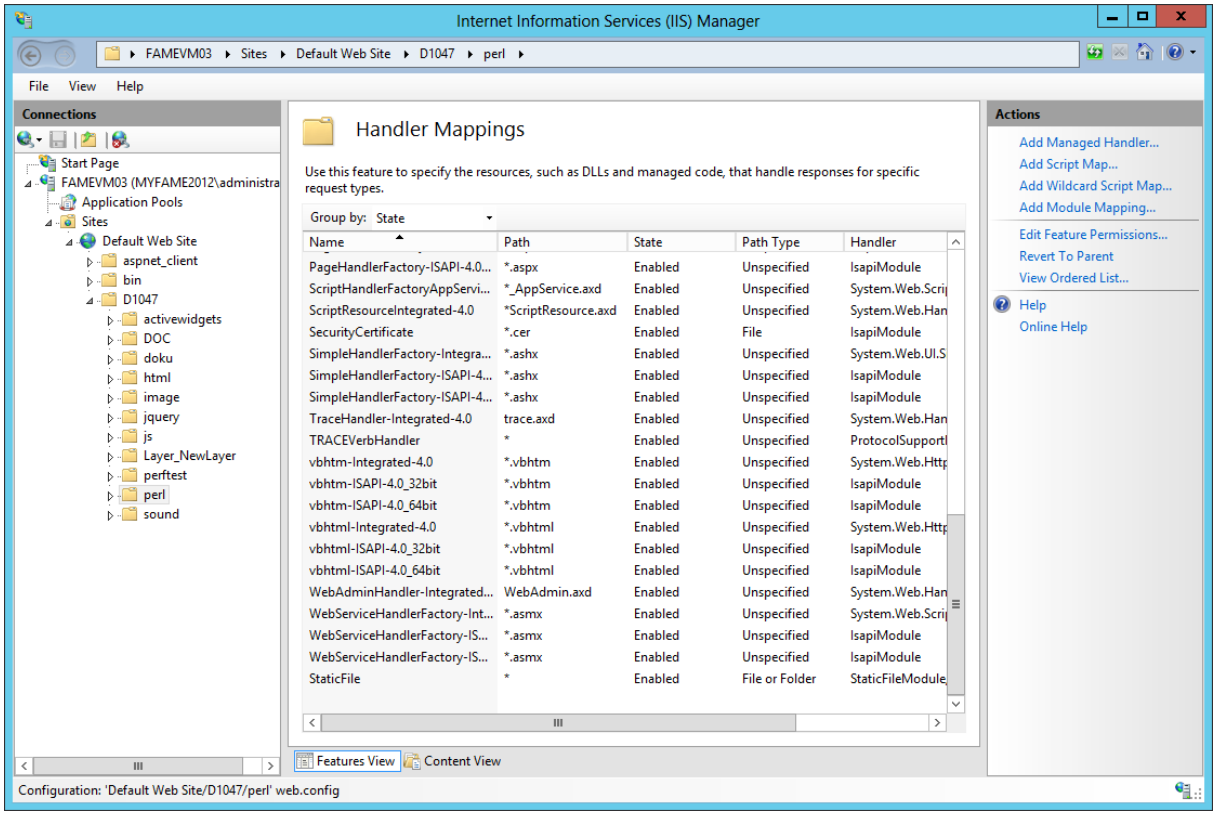
2.9.13.1 IIS configuration for Perl using IIS Manager

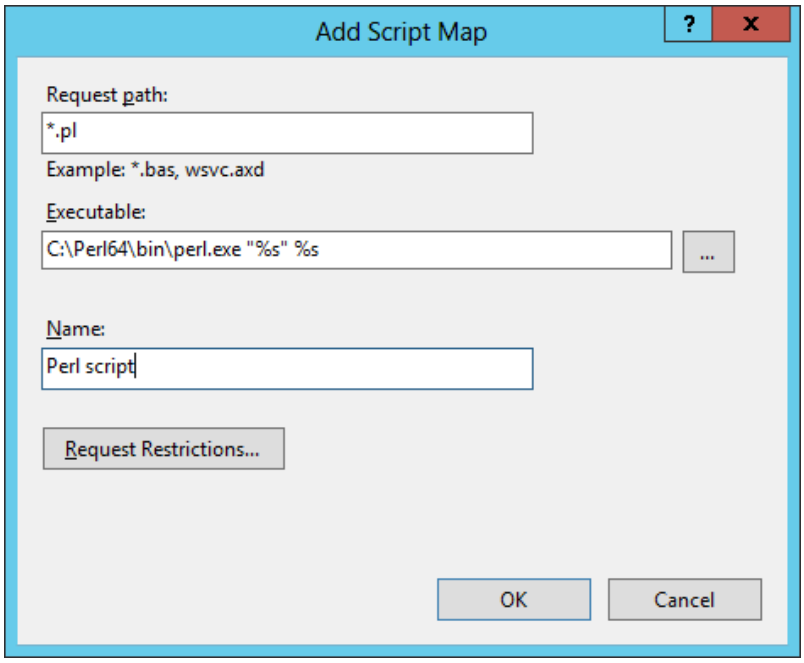
To enable the execution of Perl scripts by the IIS a handler mapping has to be added.

Select the `perl` folder of the application and open the Handler Mappings feature:

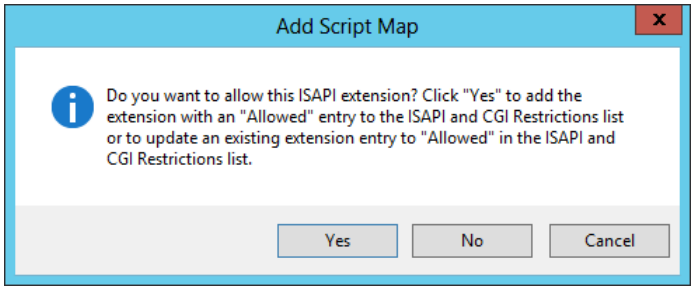


Click *Add Script Map* and enter the configuration for Perl:





The double quotes around the 1st `"%s"` argument are important and must not be left out.



2.9.13.2 Authentication setting for the perl folder

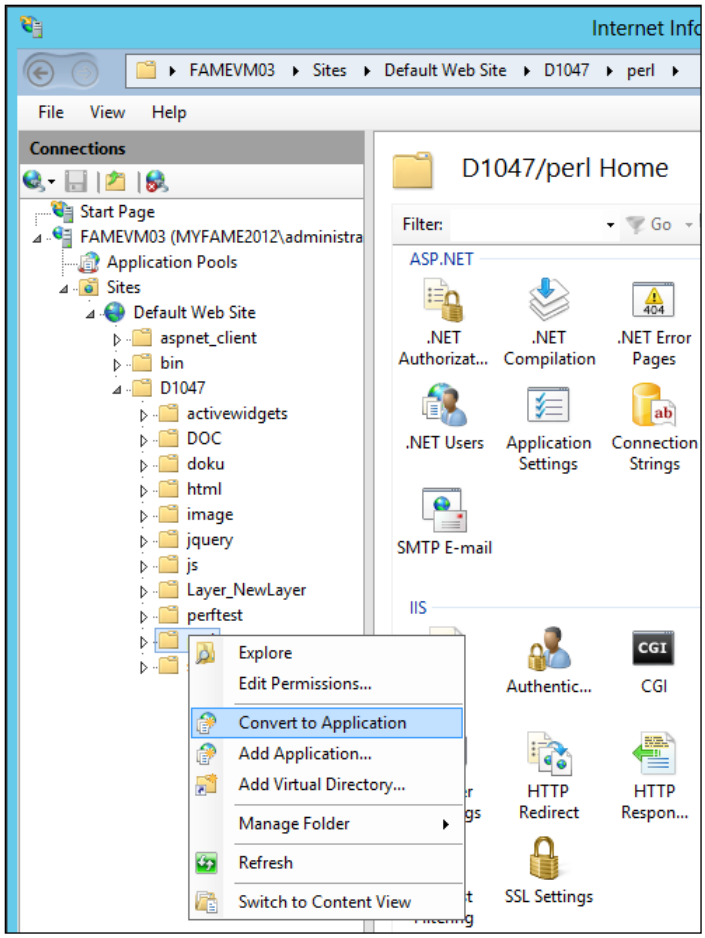
The Perl programs should run as an application pool user. Integrated Windows authentication would complicate things here as the Perl programs write and delete logs and temporary files.

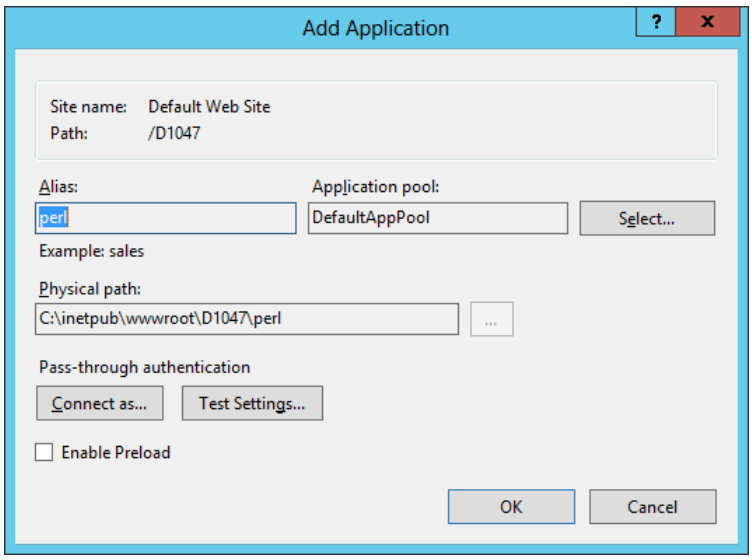
For this reason anonymous authentication is recommended for the *perl* folder.

2.9.13.3 Conversion of the perl folder to an application

The perl folder shall be converted to an application. This provides

- starting of the Perl interpreter with the correct current directory
- ability to choose a specific Windows user to run Perl scripts



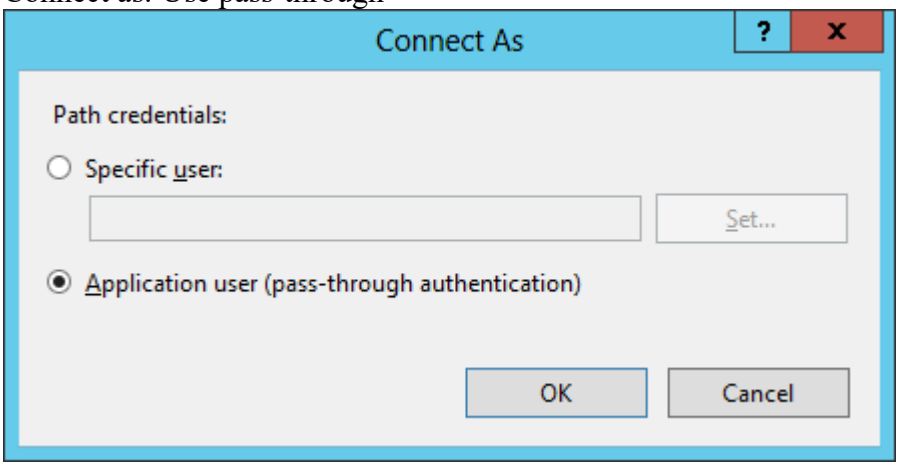


In the *Add Application* dialog the values shown have been preset automatically.

Settings:

- Alias: use the default alias set by this dialog

- Physical path: use the default path set by this dialog
- Application pool: the default settings may be used. Please note that the application pool user requires read, write and delete permissions on the folder *perl\logs*.
- Connect as: Use pass-through



The Windows user setting for the *perl* application folder is important for user permission requirements on the PDF Writer's log directory *perl\logs*. The Windows user choosen requires read and write access as the PDF Writer will create log files as well as temporary files in its log directory, and each time the PDF Writer is started it will try to delete old logfiles.

If the setting *Application user* is used together with anonymous authentication for the *perl* folder *only the application pool identity* will need to have read and write access to the log directory.

If the setting *Application user* is used together with integrated Windows authentication for the *perl* folder *all application users* will need to have read and write access to the log directory.

The log directory is configured as `log_directory` parameter in the file *pdf_report.conf*. See 2.9.11 Basic PDF Writer configuration.

2.9.13.4 IIS configuration for Perl in web.config

Alternately, the web.config file can be changed to set up the configuration for Perl scripts:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <handlers>
            <add name="Perl script" path="*.pl" verb="*"
modules="CgiModule" scriptProcessor="C:\Perl64\bin\perl.exe
&amp;quot;%s&amp;quot; %s" resourceType="File" />
        </handlers>
    </system.webServer>
</configuration>
```

This very short file was actually created using the IIS manager inside the `perl` folder itself.

The handler configuration may be integrated into the `<handlers>` section of the web.config file in a superior level directory as the FaMe application does not use any Perl script outside of the `perl` folder.

This configuration does not contain the conversion of the `perl` directory to an application, so that step should be done using the IIS manager.

2.9.14 Testing the FaMe PDF Writer

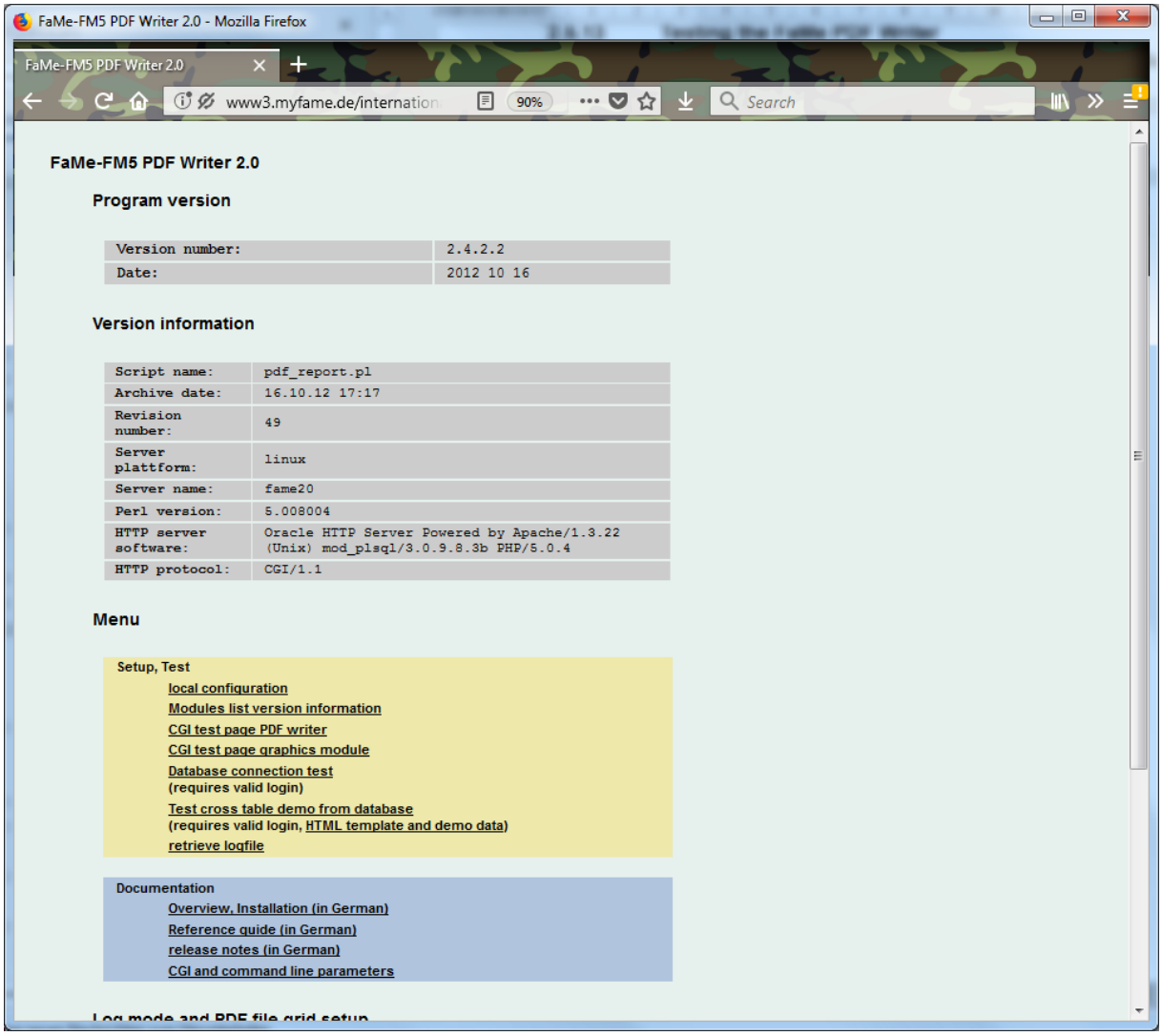
To test the installation of the FaMe PDF Writer log in as an administrative FaMe user. Manually change the URL to

```
http://myserver/application/perl/pdf_report.pl
```

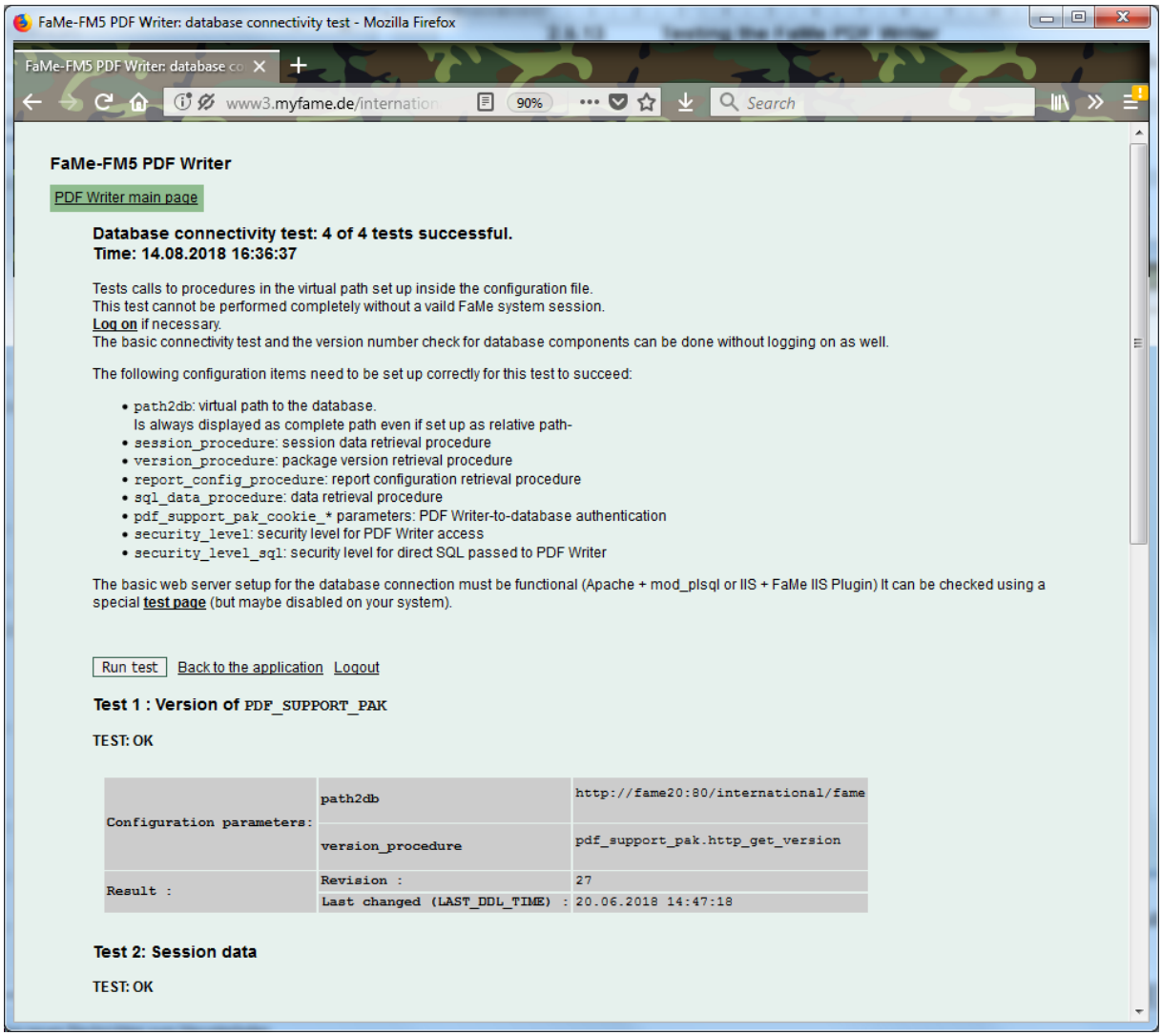
If your application is inside the virtual path /D1047, the PDF Writer will be in

```
http://myserver/D1047/perl/pdf_report.pl
```

The following page should be shown:

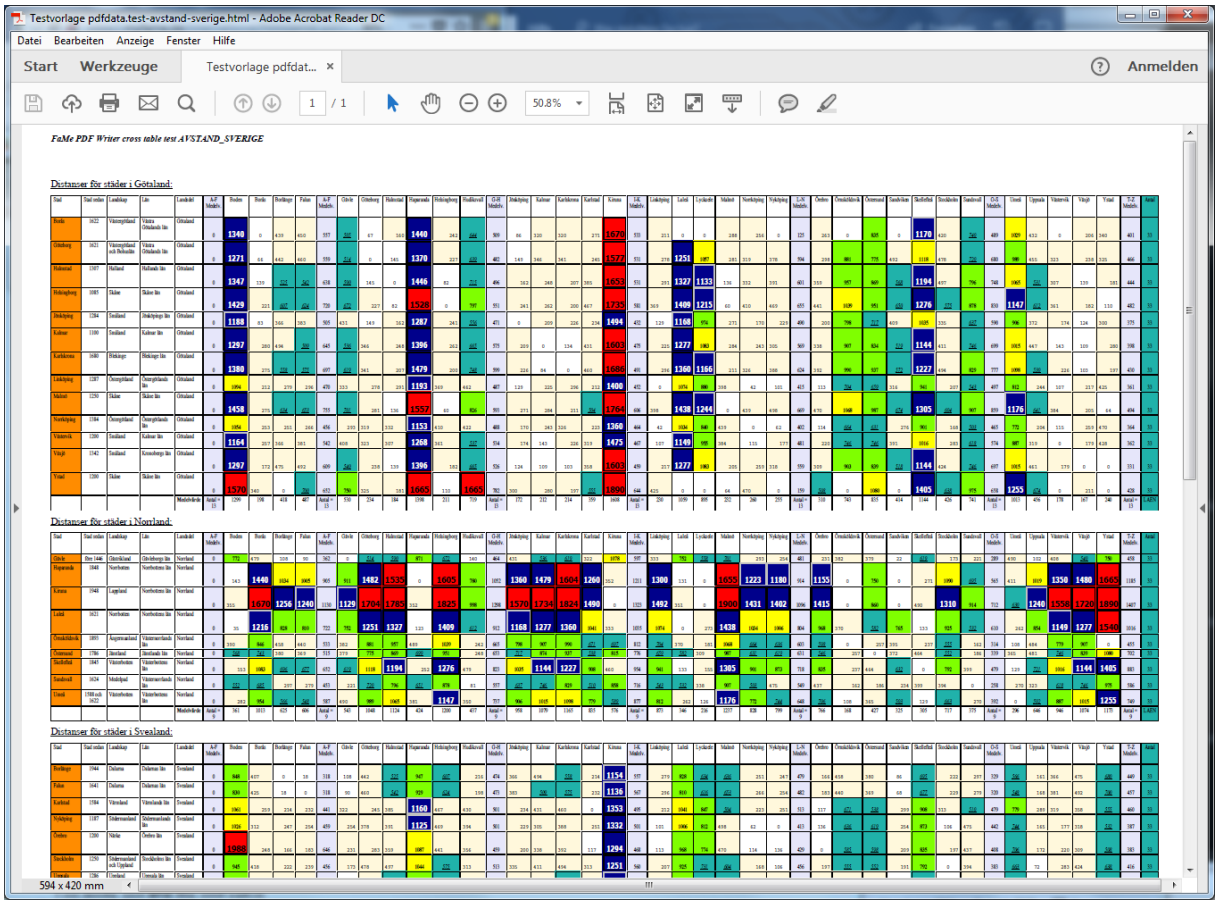


On this page click onto the link *database connection test*. The resulting page will show the results:



The first 3 of these tests must succeed while Test 4 may fail for reasons that are not caused by the PDF writer.

If the first 3 of these tests where successful return to the menu page and click onto the link *Test cross table from database*. A popup should open now and display the results of a demo PDF that is based on a special table with test data:



This test requires the template file *pdfdata.test-avstand-sverige.html* to be installed in the directory configured as *disk_template_directory* in the *pdf_report.conf* file (or one of its subdirectories).

2.9.15 PDF Writer troubleshooting

This paragraph is identical with the text file FaMe-FM5.pdf_report_generator.troubleshooting.en.Rev.0.txt.

This section covers the following errors situations:

- PDF Writer error 1009
- PDF Writer error 1301
- HTTP-403/Forbidden
- HTTP-500 Internal Server error
- HTTP-502 Internal Server error
- "You do not have permission to access this URL."

2.9.15.1 PDF Writer error 1009

Case 1: HTTP-403 on the database side

```
FaMe-FM5 PDF Writer
In the production of the PDF document, an error has occurred.
Error Number: 1009
text:
Error during HTTP access to the URL
http://server/path/fame/<procedurename>
>>> Error: 403 Forbidden. <<<
```

Possible causes:

- The current PDF-Writer security settings forbid the access
- The current PDF-Writer security settings are out of sync with the PDF writer security settings inside the database
- The use of the PDF Writer cookie in communication with the database is enabled, but the system clocks between the web server and database server differ too much

Case 2: HTTP-500 on the database side

```
FaMe-FM5 PDF Writer
In the production of the PDF document, an error has occurred.
Error Number: 1009
text:
Error during HTTP access to the URL
http://server/path/fame/<procedurename>
>>> Error: 500 Internal Server Error <<<
```

There was an error calling a database-side PL/SQL procedure. The called procedure or package may have an invalid compilation status, or a runtime error occurred inside the database.

Case 3: HTTP-500 on the database side

```
FaMe-FM5 PDF Writer
In the production of the PDF document, an error has occurred.
Error Number: 1009
text:
Error during HTTP access to the URL
http://server/path/fame/<procedurename>
>>> Error: 503 Service Temporarily Unavailable <<<
```

The database is down or not available via the webserver.

2.9.15.2 PDF Writer error 1301

This error may be caused by

- A misconfiguration of a report or printing function. Contact FaMe support in this case.
- The *security_level_sql* setting inside the `pdf_report.conf` file does now allow the PDF Writer to accept SQL statements from the client.
  A setting of ADMIN restricts this to administrative FaMe users (these are *not* administrative Windows users).
  A setting of MAX entirely disables the evaluation of client-side SQL statements.
  Usually, *security_level_sql=MAX* is the recommended setting.

2.9.15.3 HTTP-500 Internal Server error

The execution of the Perl interpreter or execution of the Perl script `pdf_report.pl` failed. Check the IIS configuration and the path to the Perl interpreter. Additionally, you might try to run the script `pdf_report.pl` from the command line:

```
cd \inetpub\wwwroot\path2app\perl
perl pdf_report.pl --help
```

This just prints the online help. Try this as well:

```
perl pdf_report.pl --list-all-libs
```

If there is any problem with the Perl and/ or library installation there should be an error message.

2.9.15.4 HTTP-502 Internal Server error

The HTTP-502 error is a special error thrown by IIS of the CGI program started prints messages to STDERR. It may be necessary to use the script `start_pdf_report.pl` that is explained above: *2.9.7.3 Script pdf_report.pl*.

2.9.15.5 "You do not have permission to access this URL."

This is a message by the FaMe PDF Writer returned when there is no valid FaMe user session. Log in to FaMe and try again.

2.10 Setting up the FaMe Graphics module

The FaMe Graphics Module provides several types of pie, bar, and line charts plus a few special forms like panel meter-like displays. It is an optional extension to the FaMe application and is basically an interface to the *Perl Chartdirector* library by *Advanced Software Engineering Ltd*.

2.10.1 Requirements

The FaMe Graphics module requires

- The FaMe Perl library
- The *Perl Chartdirector* library
- The script *fm5_chart.pl*
- For IIS only: the script *start_fm5_chart.pl*
- The configuration file *fm5_chart.conf*
- An Oracle client
- For testing the FaMe PDF Writer must be installed as well as it provides the test page

Note: the FaMe PDF Writer provides the test page, but the PDFlib library required to print PDFs is not used by the Graphics module.

2.10.2 The Perl Chartdirector library

The Perl Chartdirector library is available for download at
https://www.advsofteng.com/download.html

On that page scroll down to the *ChartDirector for Perl* section and choose the version for 64 Bit Windows. It is the file *chartdir_perl_win64.zip*.

To install it, extract the zip file and copy these files into the site\lib folder of the Perl environment. This is the same folder into which the FaMe Perl library has been copied before.

Files to copy:

- ChartDirector\lib*.dll
- ChartDirector\lib\perlchartdir.pm

The screenshot shown in paragraph *2.9.7.1 FaMe Perl Library* shows the location where these files must be copied into.

2.10.3 Scripts fm5_chart.pl and start_fm5_chart.pl

The script `fm5_chart.pl` is the FaMe Graphics module. It uses the FaMe Perl library and the Perl ChartDirector library. Configuration is in file `fm5_chart.conf`.

On Windows, a special script `start_fm5_chart.pl` is required to start this script as it prints to STDERR, causing the IIS to report a HTTP-502 error. To actually use it, it must be renamed to `fm5_chart.pl` and `start_fm5_chart.pl` must be renamed to `s_fm5_chart.pl`:

- `fm5_chart.pl => s_fm5_chart.pl`
- `start_fm5_chart.pl => fm5_chart.pl`

Before renaming check the file contents. Script `start_fm5_chart.pl` is very short and contains the line

```
if (system("C:\\Perl64\\bin\\perl.exe s_fm5_chart.pl 2>NUL") != 0) {
```

while the original `fm5_chart.pl` script contains this Source Safe header:

```
# $Header: /5068.PDF-Writer/cgi/fm5_chart.pl 2 16.10.12 17:04 Elwood $
# $Revision: 2 $
# $Date: 16.10.12 17:04 $
```

Both scripts are Perl scripts and require the IIS to be configured to run them using the Perl interpreter. The configuration is described above for the FaMe PDF Writer. See *2.9.13 IIS configuration for Perl scripts*. If the FaMe PDF Writer setup is already done no extra IIS configuration is required.

The script requires an Oracle client as well. As long as an Oracle client is already in place as required by the FaMe IIS plugin no extra action needs to be taken.

The Oracle client's version is uncritical as the interface used is the Perl modules *DBI* plus *DBD::Oracle* which internally uses the Oracle Call Interface (OCI).

2.10.4 Configuration file fm5_chart.conf

The FaMe Graphics Module is configured using the file *fm5_chart.conf.*

It is a plain text file. Comments start with a pound sign '#'.

The parameters are described here, and the important ones that must be checked are marked *YES* in the *Check* column:

Parameter	*Description*	*Check*	*Standard value*
tns_hostname	Either a real TNs hostname that is configured inside the *tnsnames.ora* file or an Oracle "*EZ configuration*" string: server:port/Instance e.g. famedbserver:1521/FAMEDB	YES	None, setup this up manually
db_username	Database username	YES	Always FAME_WEB
db_password	Database password	YES	
db_charset	The character set to use, ISO form, not Oracle form. Typically UTF-8	YES	UTF-8
env.NLS_LANG	The NLS_LANG variable to use by the	YES	AMERICAN_AMERICA.AL32UTF8

	Oracle client		
env.FONTPATH	The path to the font files If the standard Windows fonts are used this parameter may be commented		C:\windows\fonts
logfile	The name of the logfile, may contain patterns for date and time Compare the pattern used by the POSIX function *strftime* for more information		fm5_chart.%y%m%d.log
log_mode	The log mode. Choices are none, log, verbose, debug		None
log_dir	The log directory to use. Must be writable by all users this module runs as. Compare section *2.9.12 The log directory for the FaMe PDF Writer*.	YES	Logs

2.11 Testing the FaMe Graphics Module

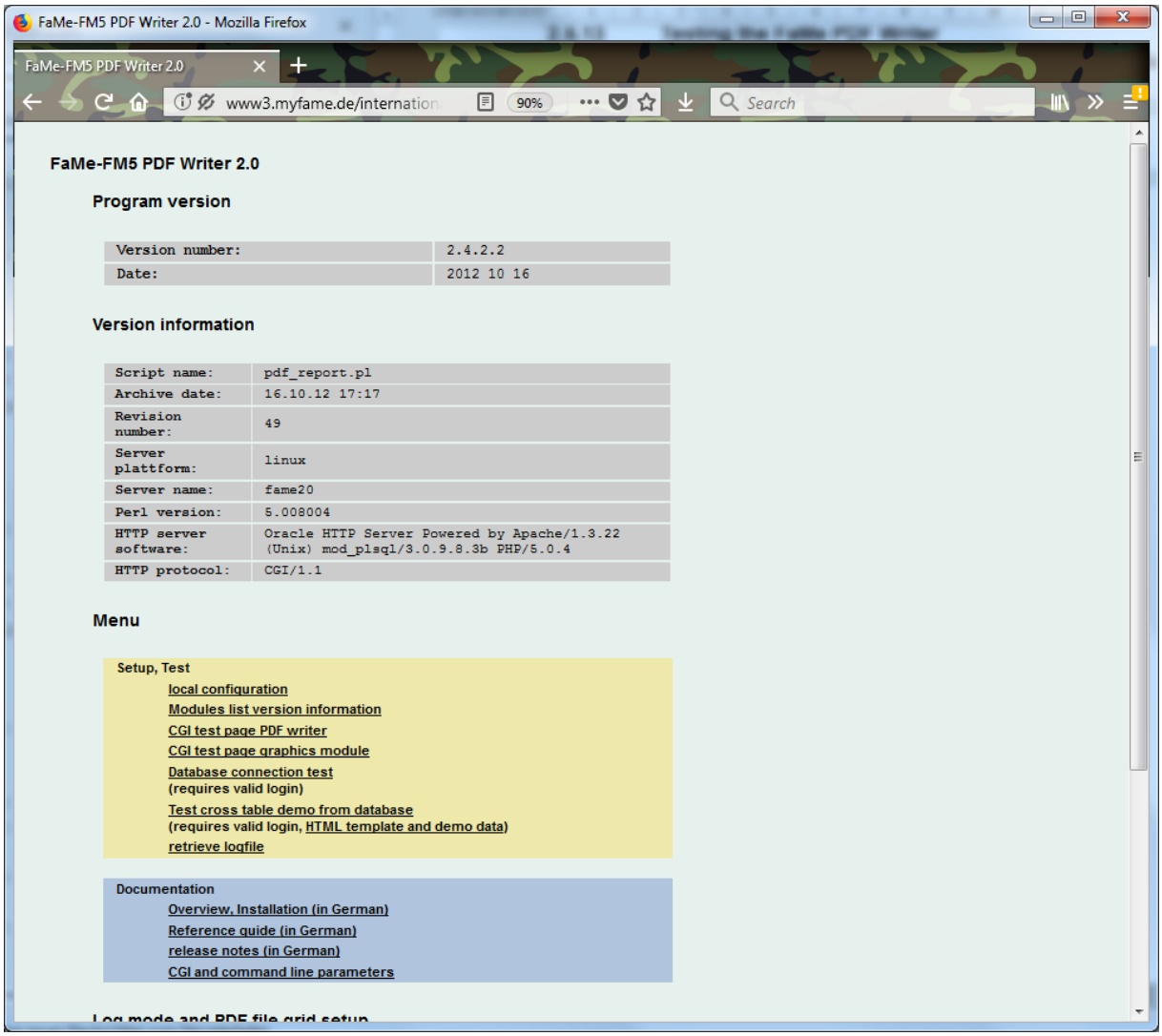
To test the installation of the FaMe PDF Writer, log in as an administrative FaMe user. Manually change the URL to

```
http://myserver/application/perl/pdf_report.pl
```

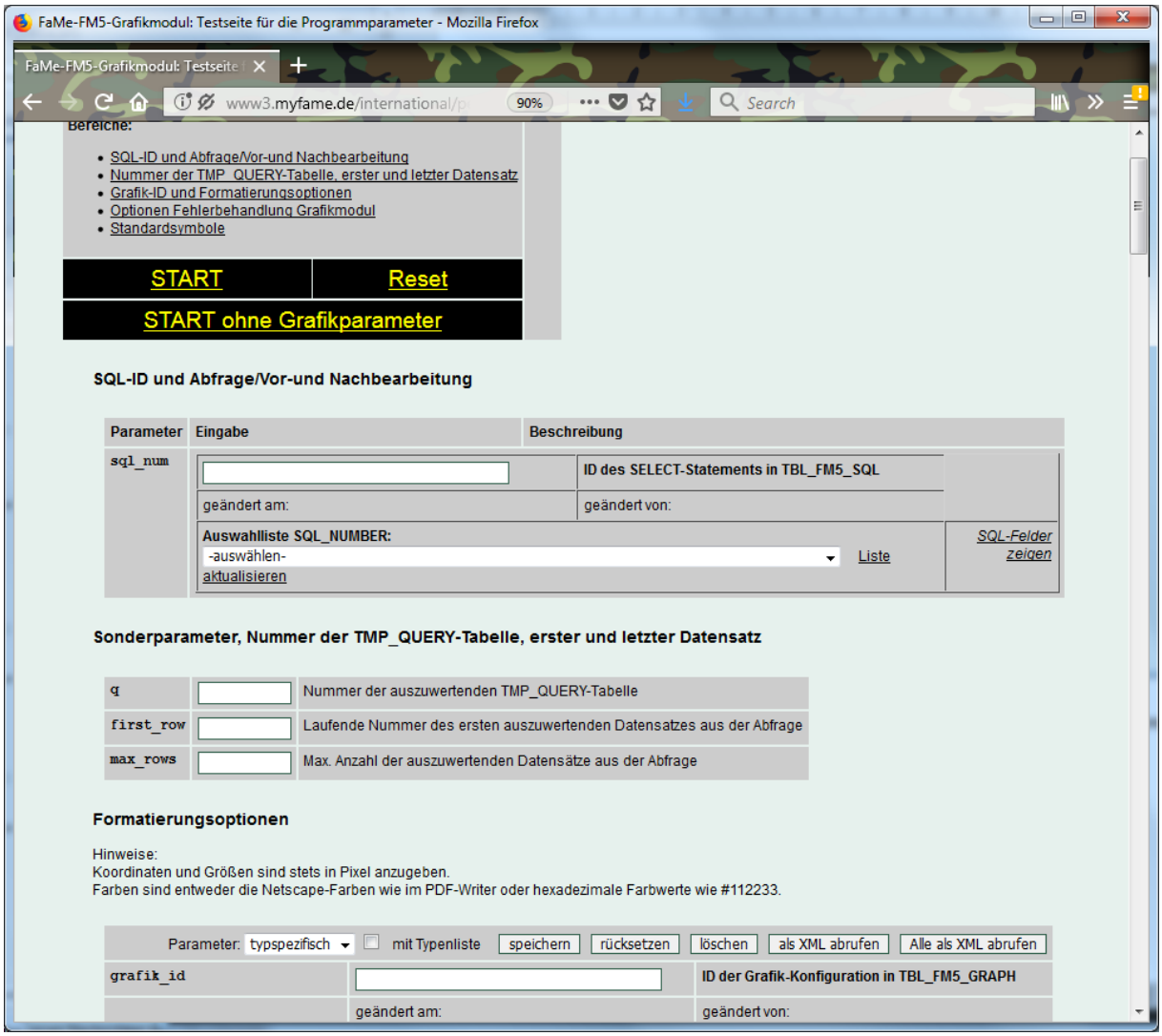
If your application is inside the virtual path /D1047, the PDF Writer will be in

```
http://myserver/D1047/perl/pdf_report.pl
```

The following page should be shown:



On this page click onto the link *CGI test page graphics module*. The resulting page will be shown:

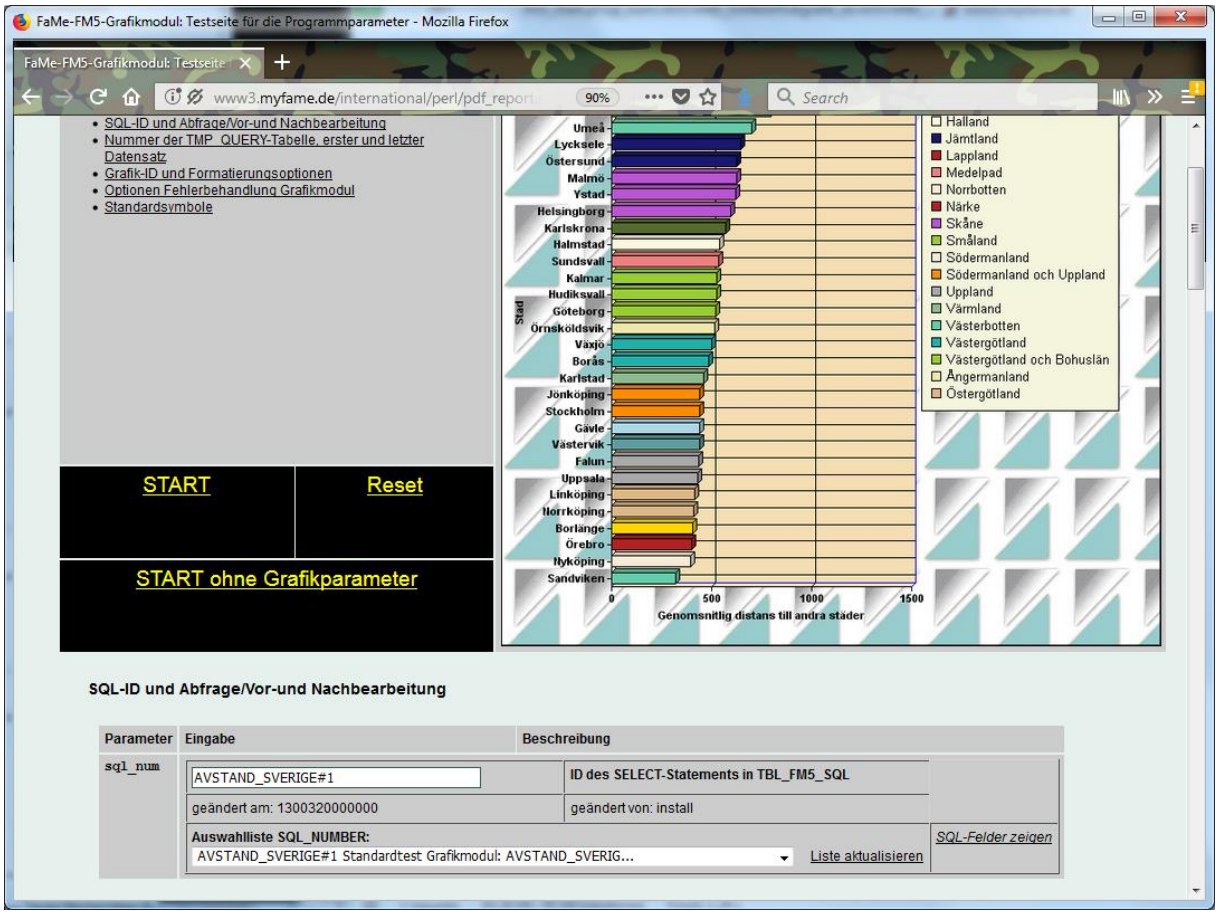


(There is currently no translation available for this page)

On this page choose

- Auswahlliste SQL_NUMBER: AVSTAND_SVERIGE#1
- Auswahlliste Grafik-ID: AVSTAND_SVERIGE#1

Click the *Start* link to start the test. The result should look like this:



If this result is not shown look for the parameter *image_type* on the page. If this is set to *svg*, try again with *png* or *jpg*.

FaMe Facilities Management Software GmbH

An der Autobahn 43 a

D - 28876 Oyten
Germany

Phone: +49 4 207 / 91 10-0
Fax: +49 4 207 / 91 10-19

https://www.fame-online.de