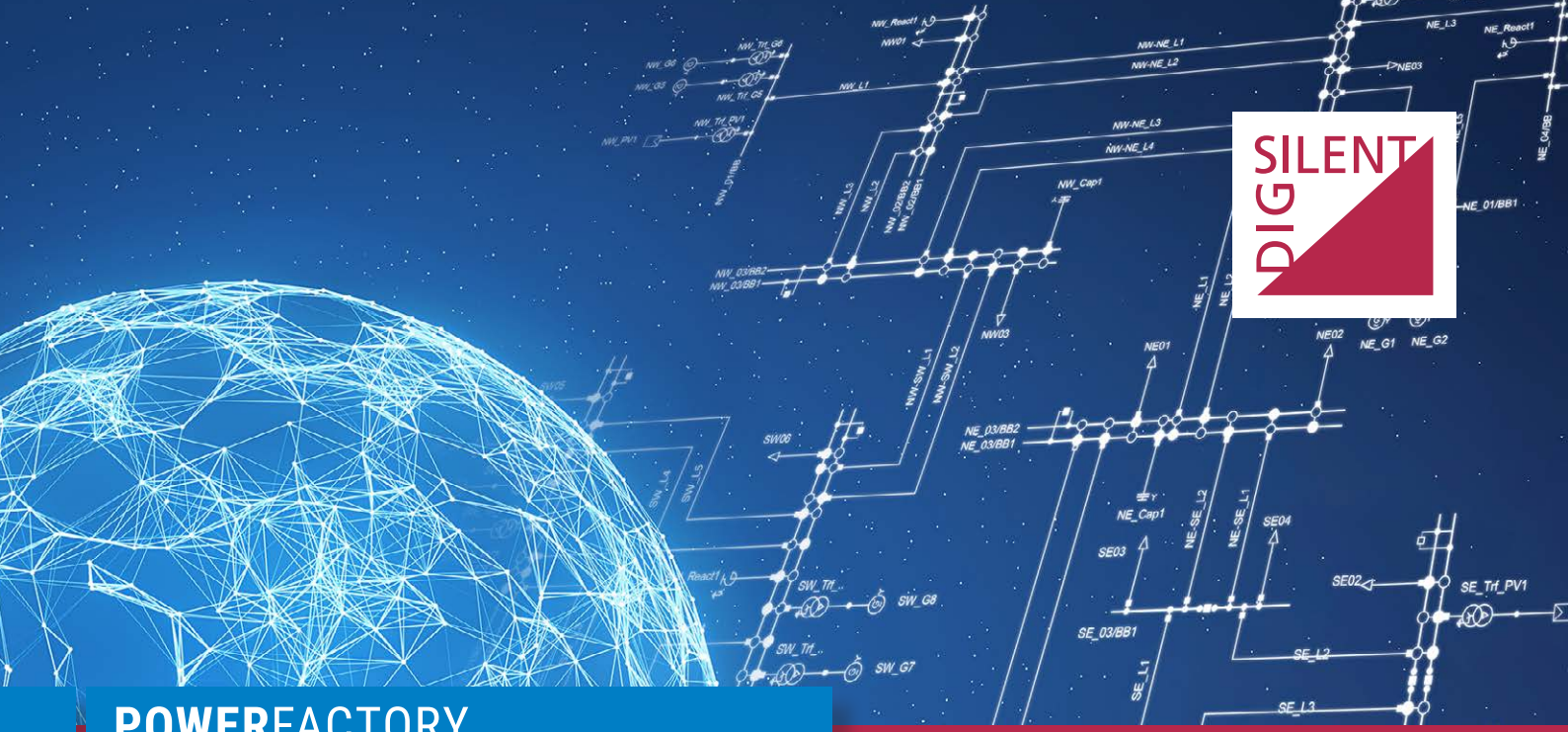

POWERFACTORY

PowerFactory 2021

Advanced Installation and Configuration Manual

PF2021

POWER SYSTEM SOLUTIONS

MADE IN GERMANY

Publisher:
DIgSILENT GmbH
Heinrich-Hertz-Straße 9
72810 Gomaringen / Germany
Tel.: +49 (0) 7072-9168-0
Fax: +49 (0) 7072-9168-88
info@digsilent.de

Please visit our homepage at:
https://www.digsilent.de

Copyright © 2020 DIgSILENT GmbH
All rights reserved. No part of this publication may be reproduced or distributed in any form without written permission of DIgSILENT GmbH.

December 14, 2020
PowerFactory 2021
Revision 1

Contents

Chapter 1

Introduction

1.1 Overview

Depending on the installation type, a *PowerFactory* system can have several components which have to be installed and configured separately.

- The separately available "Getting Started" document covers the basic installation options.
- More advanced installation options e.g. multi-user database, installation on an application server, and the Offline mode installation are described in this document.

1.2 *DIgSILENT* Download Area

Additional Software, Documents, and Examples for *PowerFactory* can be downloaded from the Download Area on the *DIgSILENT* website https://www.digsilent.de/en/downloads.html.

Please note that access to the Download Area is granted for registered users only. The user registration can be done via the support page https://www.digsilent.de/en/user-registration.html. It initially requires the input of the company credentials that can be found in the licence agreement document.

1.3 Knowledge Base of Frequently Asked Questions (FAQ)

A “Knowledge Base” database of information for commonly appearing issues, is available for any users (whether registered or not) at https://www.digsilent.de/en/faq-powerfactory.html

1.4 Technical Major Changes

1.4.1 *PowerFactory* 2021

No changes.

1.4.2 *PowerFactory* 2020

- Support for Oracle Client versions 18c and 19c.

1.4.3 *PowerFactory* 2019

- Support for Oracle Client 12c Release 2.
- New Security & Privacy features: Audit Log (see section 7.1 on page 39), Idle Session Timeout (see section 7.2 on page 42), and Privacy settings (see section 7.3 on page 42).
- The application is built with Visual Studio 2017. Appropriate runtimes are shipped with the installer
- A new setup assistant guiding through initial configuration is now provided.
- The maintenance period is now stored on the *PowerFactory* licence. This means a licence update will be required whenever the corresponding contract is prolonged.

1.4.4 *PowerFactory* 2017

- Due to *PowerFactory*'s dependence on CodeMeter, the .NET 4.5.1 runtime and the Visual C++ Redistributable, the installer package has been further extended and now uses the common bootstrapper approach. Its format has therefore changed to a standard executable.
- Oracle ODBC client is now supported (see section 6.2.2.5 on page 29).
- It might be necessary to update existing licences in order to access full functionality of *PowerFactory* 2017. At the end of the installation process, the ***Licence Manager*** will be launched allowing for an online search for updates (see section 5.4 on page 12 for more details on licence updates).
- A new Upgrade Assistant allows to transfer the applications settings (e.g. workspace directory, database settings...) of an already existing installation. The older installation must be from version 2016 or higher to be found by the Upgrade Assistant.

1.4.5 *PowerFactory* 2016

- *PowerFactory* now uses the Windows Installer Engine and is therefore shipped as MSI package.
- The installation procedure has been greatly simplified.
- The whole licensing is now based on WIBU CodeMeter technology.

Chapter 2

PowerFactory Editions Overview

PowerFactory is licensed in three different editions:

The *Single User Edition* is intended to be installed on a workstation with a local dongle or softkey.

Figure 2.0.1: Single User Edition

The *Multi User Edition* allows for a dedicated licence server to be used. Such a licence server provides easy licence sharing and concurrent work for multiple users[1] within same local network.

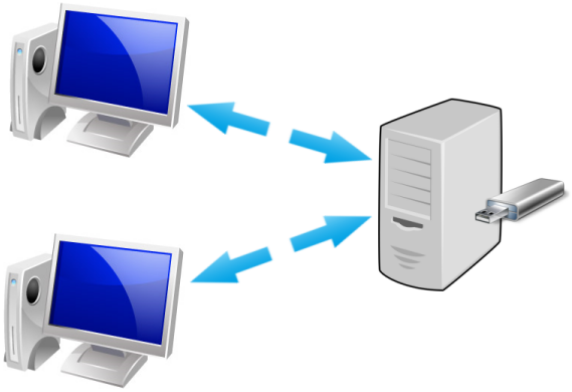

Figure 2.0.2: Multi User Edition

[1] Depending on the number of licensed concurrent users.

The *Team Edition* extends the *PowerFactory* system architecture with a database server for centralised storage. It consists of a central licence server and allows usage of a central multi-user database.

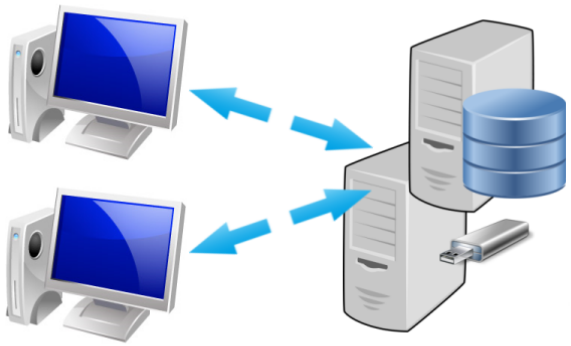

Figure 2.0.3: Team Edition

The following table gives an overview of components that need to be installed depending on the purchased *PowerFactory* edition:

	Single User Edition	**Multi User Edition**	**Team Edition**
PowerFactory Installer (on all client computer(s))	X	X	X
LicenceComponentsBundle Installer (on central licence server computer)	—	X	X
Oracle or Microsoft SQL Server DBMS(*) (on central database server)	—	—	X

Table 2.0.1: Installation Components

(*) Not provided by *DIgSILENT*

Chapter 3

System Requirements

3.1 *PowerFactory*

DIgSILENT *PowerFactory* is an application for standard Windows operating systems. Both a 32-bit and a 64-bit version are available. For running *PowerFactory* efficiently the computer should be equipped with, as a minimum:

- Operating system:
  - Windows 8.1 or Windows 10
  - Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, or Windows Server 2019
  - .NET Framework 4.5
  - The 64-bit version of *PowerFactory* requires a 64-bit operating system
  - The 32-bit version will run on both architectures

  Note: Installation on Windows 8.1 requires KB2999226. To install through Windows Update, make sure you install the latest recommended updates and patches from Microsoft Update before you install the Windows SDK.
- Processor with 2 GHz
- Main memory of 2 GB RAM
- Hard disk space of 2 GB plus additional 5 GB per user
- SVGA graphic card with a resolution of at least 1280x1024 pixels

Additional requirements:

- Internet connection for licence activation, transfer and regular licence validity checks (every 30 days)
- Administrator privileges for the installation process

Optional requirements for data export / import:

- MS Excel 2010 or newer
- MS Access 2010 or newer with ODBC drivers with the same architecture (32 or 64 bit) as *PowerFactory*

Optional requirements for Multi-User Database

- Microsoft SQL Server Database
  - SQL Server 2012, 2016, 2017 or 2019

  - ODBC drivers for SQL Server (typically shipped with the operating system)

- Oracle Database
  - Oracle Client 19c, 18c, 12c Release 2, or 12c Release 1 with the same architecture (32 or 64 bit) as *PowerFactory* (both full client and Instantclient are supported)
  - Oracle server
    - Oracle Client 19c supports: Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
    - Oracle Client 18c supports: Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
    - Oracle Client 12c Release 2 supports: Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
    - Oracle Client 12c Release 1 supports: Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
- Fast network (100 Mbit/s or higher) with low latency (<1ms) connection between the database server and the machines where *PowerFactory* is running.

Optional requirements for Python scripting:

- Python version 3.6, 3.7, 3.8 or 3.9 with the same architecture (32 or 64 bit) as *PowerFactory*

3.2 Network Licence Server (optional)

For network licences it is recommended to set-up a dedicated network server. The machine should fulfil requirements as follows:

- Dedicated server:
  - The server machine should be used exclusively for *PowerFactory*.
  - Virtualisation:
    - For softkey licences, a physical machine is required.
    - For dongle based licences (hardlock) a virtual machine can be used (in conjunction with an appropriate USB dongle server, see 5.2).
- Operating system:
  - Windows 8.1 or Windows 10
  - Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, or Windows Server 2019
  - .NET Framework 4.5
- Processor with 2 GHz
- Main Memory of 2 GB RAM

Additionally:

- Fast network with short latency (100 Mbit/s or higher) connection between the licence server and the machines where *PowerFactory* is running.
- Internet connection for licence activation, transfer and regular licence validity checks (every 30 days)
- Administrator privileges for the installation process

See also next chapter for installation and configuration.

Chapter 4

Licence Server

Running *PowerFactory* requires a valid licence. This can either be a *workstation licence* that is activated on the same machine as the *PowerFactory* installation, or a *network licence* that is granted by a licence server in your local network.

This section describes the installation and configuration process of a *PowerFactory* licence server.

Note: A licence server typically provides a network licence for a number of client machines (Multi User or Team Edition licence). Although it is possible to install a workstation licences on a server machine, this licence will not be accessible by client computers in your network.

4.1 Installation

Please run the *Licence Components Bundle* installer and follow the on-screen instructions which will guide you through the installation process. The following components are installed:

- WIBU CodeMeter Runtime Server as Windows service
- ***Licence Manager*** required for activating the network licence on the server machine
- ***Licence Validation Service*** as a windows service. The installation of this service is optional, but recommended. It performs regular online validation locally from the licence server without the need to configure online access on any *PowerFactory* client.
- Installer package for ***Legacy Licence Service***. The installation of this service is optional. It allows *PowerFactory* 15.2 or 15.1 versions to be executed with a recent *PowerFactory* licence. A separate installation is required. See 9.1.1.

4.2 Configuration

For network licences, two settings are critical:

- Correctly configured internet connection: see 5.1 for details.
- Licence server status: for a server licence to be visible in the local network, it is mandatory to start a licence server on the server machine. The installation process includes steps to automatically start the licence server. However, if problems with licence visibility occur, the licence server status can be reviewed in a separate tab of the network settings dialogue in the ***Licence Manager*** (see 4.2.1).

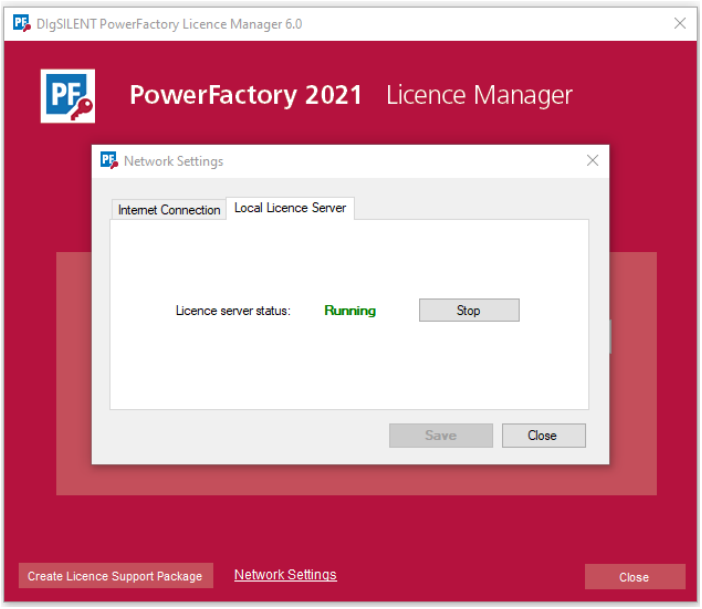

Figure 4.2.1: Licence Server Status

The handling of a network licence regarding activation, update, and moving does not differ from that of a workstation licence. Please follow the instructions given in Section 5.3 in order to activate your network licence on the server machine.

Once the licence has been activated, it should automatically be detected by all *PowerFactory* installations on computers in your local network. If a client machine has trouble to access the network licence, explicitly select the network licence on that computer as described in Section 5.5.

Chapter 5

Licence Management

PowerFactory uses the CodeMeter technology by WIBU-SYSTEMS AG for software protection and licensing. The CodeMeter runtime is implicitly installed as a Windows service by both the *PowerFactory* and the *Licence Components Bundle* installers. CodeMeter can be configured via its WebAdmin web interface: `http://localhost:22350`. Modifying CodeMeter settings directly, however, is usually not necessary and should only be done under the guidance of *DIgSILENT* support.

The ***Licence Manager*** is the primary tool for handling *PowerFactory* licences. It allows the user to activate a licence on a workstation or a licence server, to update an already installed licence, and to move a softkey licence to another computer. Furthermore, on machines with access to multiple licences, the specific licence to be used by a *PowerFactory* installation can be selected.

The ***Licence Manager*** can be started in one of the following ways:

- Opening the Windows' Start menu and running *Windows Start button* → *All Apps*→ *PowerFactory 2019*→ *LicenceManager*
- Running the `LicenceManager\LicenceManager.exe` in the *PowerFactory* installation directory
- Launching it from *PowerFactory* via the *TOOLS* → *Licence* menu

The ***Licence Manager*** presents the task selection page on startup:

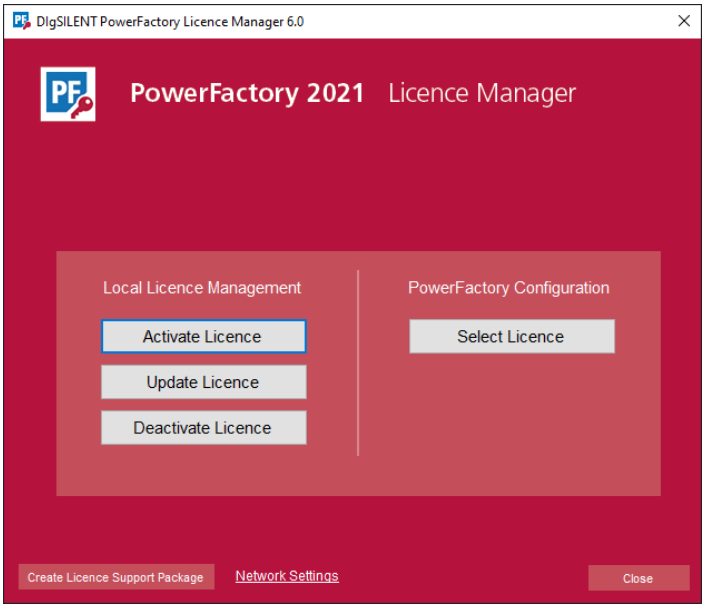

Figure 5.0.1: Startup page of the ***Licence Manager***

Note: The ***Licence Manager*** can generate a *Licence Support Package* via the respective button in the lower left corner of the program window. That package is a ZIP archive containing log files and additional diagnostic information about your system. Please include the support package when consulting the *DIgSILENT* support with licence-related issues.

5.1 Network Configuration

Online access is mandatory for all computers with local *PowerFactory* licences (Single User Edition) as well as *PowerFactory* licence servers (Multi User and Team Editions) for two different reasons:

- The licence transfer (activation, update, move) requires communication with the *DIgSILENT* server.
- Activated licences perform a periodic online check in order to verify their validity (30 day interval).

All network communication uses the HTTP/HTTPS protocol.

You can check the Internet connectivity of your machine in the *Network Settings* dialogue that is opened by the respective link at the bottom of the ***Licence Manager*** program window.

5.1.1 HTTP Proxy

If your *PowerFactory* machine (workstation or licence server) is located behind a HTTP proxy, please enter the proxy configuration in the *Network Settings*. The ***Licence Manager*** supports anonymous as well as authenticating HTTP proxies. For authenticating proxies it is recommended to use the *auto-select* authentication scheme.

The proxy settings are written to the registry and thus shared with other installations of *PowerFactory* on the same PC. In some environments it can be necessary to overwrite the proxy settings for specific installations. This can be done by manually editing the [network] section in the `PowerFactory.ini` file:

```
[network]
useHttpProxy = 1
; possible values:
; 0: none 1: system proxy 2: manual proxy 3: PAC file
httpProxyHost = myproxyserver.com
httpProxyPort = 8080
httpProxyAuth = 0
; possible values:
; 0: none 1: auto-select 2: basic 3: digest 4: NTLM 5: NTLM (Windows user) 6: Negotiate
httpProxyUser = username
httpProxyPassword = password
disableSslVerification = false
```

Note: Authenticating proxies need additionally to be configured in the WIBU CodeMeter settings. The respective configuration page of the web frontend is reachable by the URL: `http://localhost:22350/configuration/proxy.html`

5.1.2 Firewall

In case of a network licence the communication between client and licence server uses tcp/udp port 22350. This port 22350 is registered at IANA (Internet Assigned Numbers Authority) and uniquely assigned for CodeMeter communication.

Further, if your *PowerFactory* machine on which the licence resides (workstation or licence server) is located behind a firewall, please make sure it allows outgoing HTTP/HTTPS (TCP ports 80 and 443) connections from that machine to the following destinations:

- https://lc.codemeter.com/23827/gateways/
- http://cmtime.codemeter.com/

Note: Starting with CodeMeter 7.10a (delivered with *PowerFactory* 2021) there is a https alternative for the second URL. However, if you want the system to use https://cmtime.codemeter.com/ this has to be activated manually in the licence system for the moment:
open http://127.0.0.1:22350/configuration/certified_time.html and make sure that option "use https" is selected.

Figure 5.1.1: Settings page for CodeMeter licence system.

5.2 Virtual Environments

Due to copy protection reasons the activation of a softkey in a virtual environment is not supported. Nevertheless, server virtualisation is possible with a USB dongle holding a network licence. In this scenario, the USB dongle needs to be plugged either to the USB port of the physical machine hosting the VM or to a "USB via network solution" device. Then, the dongle can be connected to the VM (exclusive mode).

As the provider of the *DIgSILENT* licensing system maintains a loose cooperation with the company *SEH Computertechnik GmbH* it is recommended to use one of their "USB Dongle Server" (myUTN) solutions.

5.3 Activating a Licence

A *PowerFactory* licence is either a single-user *workstation licence* that needs to be activated on the computer where *PowerFactory* is installed, or a *network licence* that has to be activated on the machine acting as licence server. The online activation process is the same for both types of licences.

Please start the ***Licence Manager*** on the machine where you want to activate a licence and select **`Activate Licence`**. You will then be prompted to enter the "Activation Key":

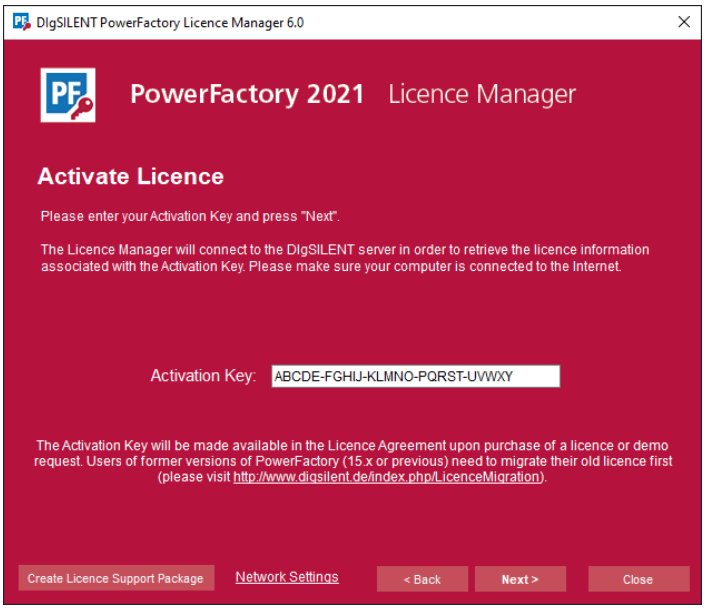

Figure 5.3.1: Online licence activation using the ***Licence Manager***

If you are a new customer, you have received your Activation Key as part of the Licence Agreement. Users of former versions of *PowerFactory* 15.x, however, need to migrate their old licence first (please visit: `https://www.digsilent.de/en/licence-migration.html`).

Please enter the Activation Key and press **`Next`**. The following page displays information about the licence associated with the entered Activation Key. If you have purchased a USB dongle licence, you are also asked to select the container you want the licence to be stored in. In this case, please plug in the USB dongle you have received with the *PowerFactory* installation package, hit the refresh button, and select the dongle in the drop-down list.

Hint: A *licence container* stores a single *PowerFactory* licence and may either be a *USB dongle* or a *softkey* that is located on a specific machine. The type of container in which a licence can reside is a property of the licence, i.e., dongle licences cannot be stored in a softkey and vice-versa. A licence container is identified by a serial of the format 123-12345678 or 3-12345678.

After clicking **`Activate`** the licence will be transferred from the *DIgSILENT* server to your computer and is then ready to be used. The activated licence should be automatically detected by *PowerFactory* on the next startup (see also 5.5 for licence selection).

5.4 Updating a Licence

There are several reasons why a customer should update a licence:

- the customer wants an upgrade of a licence, e.g. buy additional modules or increase the bus count. *DIgSILENT* sales will provide an update in this case that the customer can fetch via the update mechanism of the ***Licence Manager***.
- the customer has extended the maintenance contract of a licence. Within this process a licence update will be provided and has to be applied to the licence via the update mechanism of the

Licence Manager in order to run more recent *PowerFactory* versions (from *PowerFactory* 2019 this replaces the old mechanism of the cmApm.bin file).

- the customer installs a new major version and wants to access new functions and features that are separately licensed. In this case, the update is automatically generated and can be fetched via the update mechanism of the ***Licence Manager***.

Please start the ***Licence Manager*** on the machine where your existing licence(s) have been activated and select **`Update Licence`**.

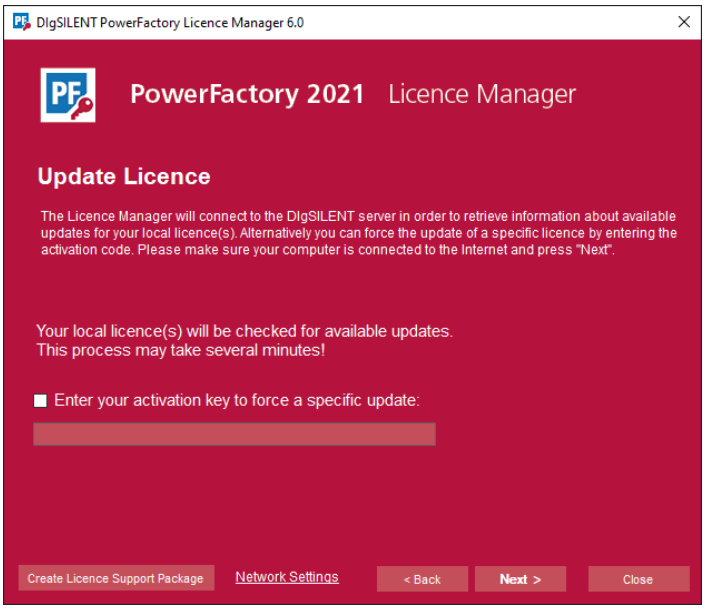

Figure 5.4.1: Online licence update using the ***Licence Manager***

Clicking **`Next`** will start an online search for updates for all your local licences. The serial numbers of the licences with an update available will be listed on the next page. You can select for each licence whether it should be updated or not.

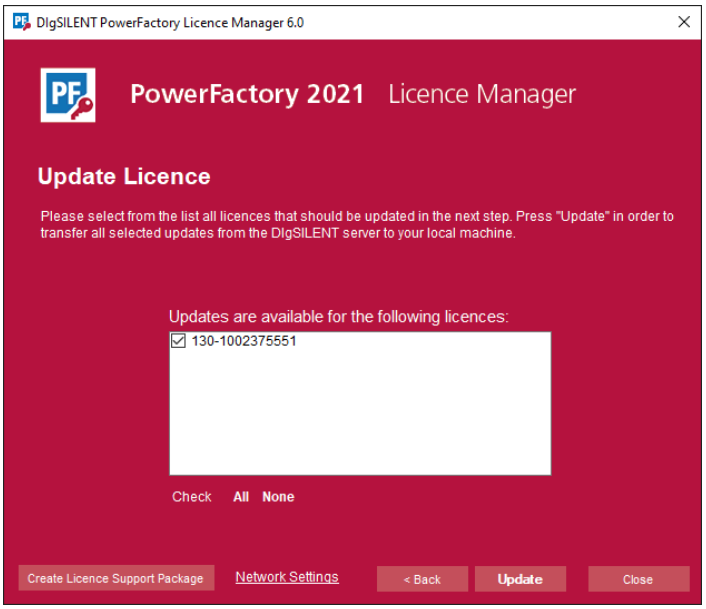

Figure 5.4.2: Selection page for updates using the ***Licence Manager***

Note: In case a customer bought a specific upgrade for one licence, he receives the activation code together with the Licence Agreement and might not want to perform an online search for update. In this case the specific activation code can be entered on the first update page. Clicking **Next** will in this case lead to a direct update of the licence under consideration.

5.5 Selecting a Licence

By default, *PowerFactory* scans the local machine and the local network for available licences and automatically picks a suitable one on startup. When there are multiple licences, or if this auto-search fails, it might however be necessary to explicitly define the licence to be used by a *PowerFactory* installation. This can be done on the **`Select Licence`** page of the ***Licence Manager***:

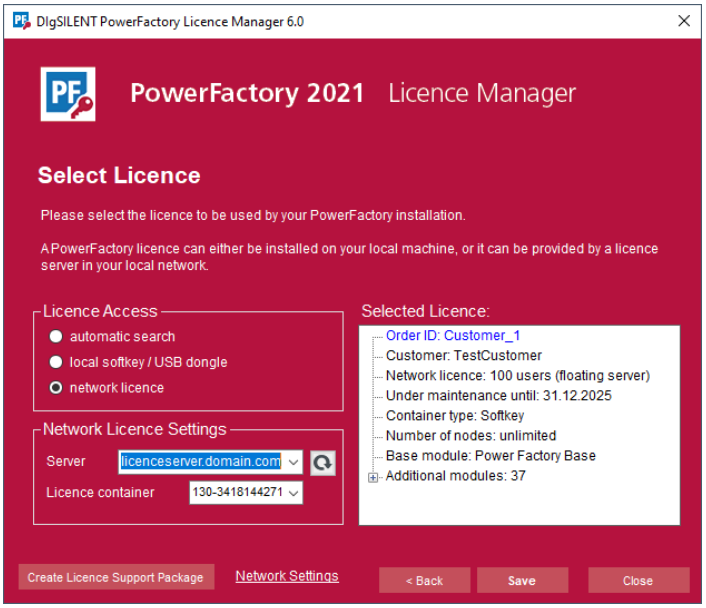

Figure 5.5.1: Licence selection page of the ***Licence Manager***

A user can select between three licence access modes:

- *automatic search*: The licence is dynamically selected by *PowerFactory* on each startup.
- *local softkey / USB dongle*: *PowerFactory* uses a specific licence that is either locally installed or stored on a USB dongle plugged into the local machine. The licence is identified by the serial of the container it is stored in. If the selected licence is not available, *PowerFactory* does not start.
- *network licence*: *PowerFactory* uses a specific network licence that is identified by the server address (hostname or IP) and container serial. If the specified licence is not available, *PowerFactory* does not start.

Hint: All available licence servers are usually listed in the drop-down list. In more complex network setups, such as virtual private networks, however, the server broadcast might fail. In this case, please enter the server address directly and hit the refresh button: If the licence server is reachable from your local machine, it should now be found.

A summary of the currently selected licence is displayed on the right of the selection page.

Please make your selection and click **`Save`**.
The licence configuration is then written to the `PowerFactory.ini` file in the *PowerFactory* installation directory, using the following section and keys:

```
[license]
container = 130-87312763
server = mylicenceserver.domain.com
```

5.6 Hot Standby Server

A network licence server grants licences to several *PowerFactory* processes running in a computer network. If this server for any reason does not respond, it will not be possible to run *PowerFactory*. The licence availability can be increased by providing a second licence (with a separate dongle / softkey) on another server within the local network. This second licence is called a *Hot Standby* licence

In general, *PowerFactory* will contact the *main* licence server. Only if the *main* licence server does not respond, the *Hot Standby* licence server will be contacted. If the *Hot Standby* licence server does not respond, *PowerFactory* will be closed with an appropriate error message.

To configure *PowerFactory* for the use of a *Hot Standby* licence, first configure the *main* licence using the ***Licence Manager*** as described in chapter 5.5. Afterwards, the `PowerFactory.ini` file has to be edited manually. In the `[licence]` section, add the keys *hotStandbyServer* and hotStandbyContainer:

```
[license]
container = 130-87312763
server = mylicenceserver.domain.com
hotStandbyContainer = 130-87309214
hotStandbyServer = myhotstandbyserver.domain.com
```

Hint: The serial number of your *Hot Standby* licence can be determined using the **`Select Licence`** page of the ***Licence Manager***: chose *network licence* and select the server holding the *Hot Standby* licence. The serial number is shown in the drop-down list.

5.7 Floating Licences

In case of using a network licence server where *PowerFactory* can only be started while connected to that server, it might be useful to have the possibility to temporarily check-out one licence and store it on the local machine allowing to work offline, i.e. detached from the network. This is generally supported with a so called *Floating Server Licence* (network licence with the *Floating Server* feature enabled (separately licensed)):

Floating Licences are time-limited local workstation licences that can be generated on demand from a *Floating Server Licence*. After generation of a *Floating Licence* only n-1 user licences will be available on the *Floating Server Licence* for the validity period of the *Floating Licence*. *Floating Licences* are typically used in *PowerFactory* offline mode (see chapter 8.4) where they are generated implicitly according to the offline mode configuration. However, it is also possible to generate, renew and return *Floating Licences* manually from within *PowerFactory*:

Generate a Floating Licence

A *Floating Licence* can be generated from *PowerFactory* via the *TOOLS* → *Licence* menu. To generate a *Floating Licence PowerFactory* must be configured to use a *Floating Server Licence*. During the generation procedure *PowerFactory* will terminate, generate the *Floating Licence* and adapt its configuration to use the newly generated licence. *PowerFactory* will use this licence from the next start.

Renew a Floating Licence

A *Floating Licence* can be generated for a maximum of 30 days. When getting close to expiration, the *Floating Licence* can be renewed to extend its validity. The renewal option can be found in the

PowerFactory TOOLS → *Licence* menu. Please note, that *PowerFactory* has to be able to reach the *Floating Server Licence* via the network to successfully perform the renewal.

Return a Floating Licence

A *Floating Licence* is time-limited and will automatically be returned to the *Floating Server Licence* after expiration (i.e. the number of licensed users on the server will be increased by 1). However, if a *Floating Licence* is no longer needed, it can manually be return via the *PowerFactory TOOLS* → *Licence* menu. Please note that *PowerFactory* has to be able to reach the *Floating Server Licence* via the network to successfully perform the return.

5.8 Hardware Changes and Softkey Licences

A *PowerFactory* software licence (softkey) is stored on the hard disk of the PC where it has been activated. Its validity is based on a hardware fingerprint of this PC. For security and copy protection reasons the softkey licence might stop working when major hardware changes are applied to the computer.

In order to avoid downtime it is recommended to deactivate a softkey licence before applying major changes to the hardware or operating system of the PC hosting the licence. After all changes have been applied, the softkey licence can be activated again on the same PC. For details on licence deactivation please refer to section 5.9.

If a PC has to be reset due to a crash or hardware failure the softkey licence might have to be restored. In this case, please contact the *DIgSILENT* support team who can guide you through the restoration process.

5.9 Moving a Licence (Softkey only)

A *PowerFactory* software licence (softkey) can be moved between computers a limited number of times per year. This can be useful, for example, in the case of hardware replacement (see also 5.8).
The licence move is a two-stage process:

1. An activated licence needs to be transferred back to the *DIgSILENT* server via the **`Deactivate Licence`** feature of the ***Licence Manager***.
2. The deactivated licence can be activated again on any computer as described in Section 5.3.

For initiating a licence move, please start the ***Licence Manager*** and click on **`Deactivate Licence`** on the startup page. The ***Licence Manager*** will then scan your computer for local licences and prompt you to select the one you want to deactivate:

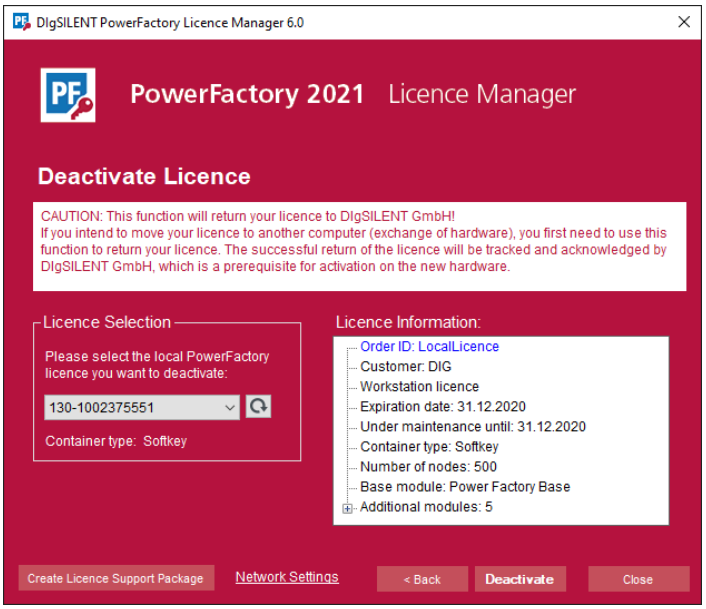

Figure 5.9.1: Licence deactivation as first step of a licence move

After you have made your selection and pressed **`Deactivate`**, you will get notified about the remaining number of moves that are allowed for the selected licence in the current year. If you confirm the deactivation, the ***Licence Manager*** will return the licence to the *DIgSILENT* server. After the successful deactivation, the ***Licence Manager*** will display the Activation Key to use for re-activating the licence on another machine.

Chapter 6

Database

6.1 Local Database

6.1.1 Workspace and Backup

PowerFactory stores data in a workspace directory in the Windows user profile. When a user named *Frodo* runs *PowerFactory*, the application data is stored usually in e.g.

```
C:\Users\Frodo\AppData\Local\DIgSILENT\PowerFactory 2019\Workspace.ComHLsIb
```

The workspace directory contains

- The local database including all projects and libraries.
- Result files (e.g. results of simulation calculations)
- Log files which are very useful when analysing application problems.
- Temporary files.

6.1.1.1 Export and Import Workspace

It's possible to manipulate e.g. backup and copy Workspace directories directly. However *PowerFactory* provides functionality for saving a workspace (including all data files in all subdirectories) as a convenient `*.zip` archive which can be used as data backup. Similarly an exported workspace `*.zip` file can be easily be re-imported into the same *PowerFactory* installation (i.e. restoring a backup) or into a completely different *PowerFactory* installation on another computer (data transfer, data migration).

The workspace functions are available in the *TOOLS* → *Workspace* sub menu:

- *TOOLS* → *Workspace*→ *Show Workspace Directory*: opens a Windows Explorer showing the workspace directory.
- *TOOLS* → *Workspace*→ *Export Workspace*: packs your workspace into ZIP archive. This may take some time.
- *TOOLS* → *Workspace*→ *Import Workspace*: deletes your current workspace and replaces it with a workspace ZIP archive.

Note: It's strongly suggested to create workspace backups on a regular basis.

Note: A Workspace import replaces the current Workspace with the imported Workspace i.e. the current Workspace is completely deleted and can not be recovered.

6.1.1.2 Workspace Directory Configuration

Storing the Workspace in the Windows user profile is convenient, but it might not be suitable under some special circumstances. Customers might want to choose a different Workspace directory:

- A company-wide policy recommends that application data should be stored inside a given directory (e.g. `D:\Data`) which is part of the company-wide backup strategy.
- Several Windows users e.g. *Frodo*, *Sam*, and *Pippin* want to work on the very same local database. Though they can't work concurrently at the same time, they might work in turns. Then the workspace should be in a directory accessible by all three users.
- Several Windows users are running *PowerFactory* concurrently on an Application Server. The installation on Application Server is described in section 8.3 on page 49.

Under these circumstances it makes sense to adapt the Workspace directory, see section 8.6.3 on page 66 for details.

Note: We strongly advise not to use a network share for the Workspace directory if the local database driver is used. Using a network share under these circumstances might lead to data loss.

6.1.2 Encryption

It's possible to encrypt the whole local database with a password.

- Log on as Administrator user
- Initiate the encryption via *PowerFactory*'s main menu *ADMINISTRATION* → *Security & Privacy*→ *Local Database Encryption*
- Set the password in the configuration (section 8.6.2.1 on page 62).

The database file is encrypted with the AES-128 or AES-256 algorithm depending on the length of the password. The password might be empty, then the database is unencrypted.

6.1.3 In-memory Mode

When using a local database, *PowerFactory* can be started in a "in-memory" mode. Once the application running in that mode is terminated all database modifications are lost. This mode is intended for automation purposes. It's similar to the "read-only" mode (section 6.3 on page 37), but usually much faster.

This mode of operation is especially useful if the instances need to import data from external sources before they can perform their work. Once the instance has been terminated all the imported data is gone, preventing the accidental accumulation of data.

As the in-memory mode uses a copy of the existing local database, it is very easy to prepare a baseline of information required for the instances to do their work. Libraries, scripts, template projects etc. can all be provided in the normal local database and are then available to all in-memory clients.

As with the read-only mode (section 6.3 on page 37), multiple instances need a dedicated instance directory for their log files, temporary files, result files etc.

Usage: the in-memory mode is enabled by the command line switch `/inMemory`. An instance identifier can be set with `/instance` e.g.:

```
PowerFactory.exe /inMemory /instance INST1
```

PowerFactory uses now a `db-INST1` directory. If it's not there it's created automatically:

```
WORKSPACE
+- db                  // normal workspace data
|   +- log
|   +- tmp
|   +- ...
|
+- db-INST1            // workspace data for instance INST1
    +- log
    +- tmp
    +- ...
```

In some cases it might be convenient to provide the in-memory switch via the PowerFactory.ini configuration file, eg:

```
[database]
inMemory = true
```

Possible values are true and false.

6.2 Multi-User Database

Normally *PowerFactory* stores the user data in a *local database* (Single User Edition) on the computer where *PowerFactory* is installed. This means if multiple users want to work on the same project, one has to export it into a file and another one has to re-import the copy into his database (see figure 6.2.1).

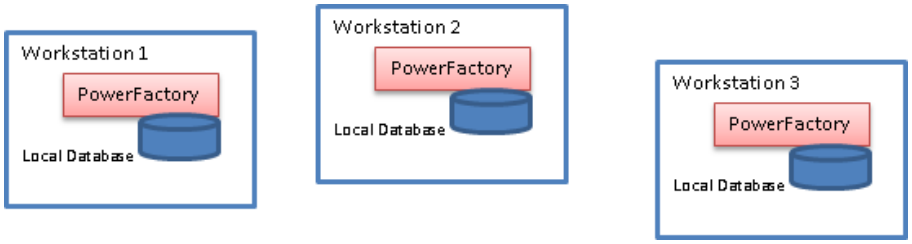

Figure 6.2.1: Local Databases

In a *multi-user database* all data is stored in one central database server (see figure 6.2.2).

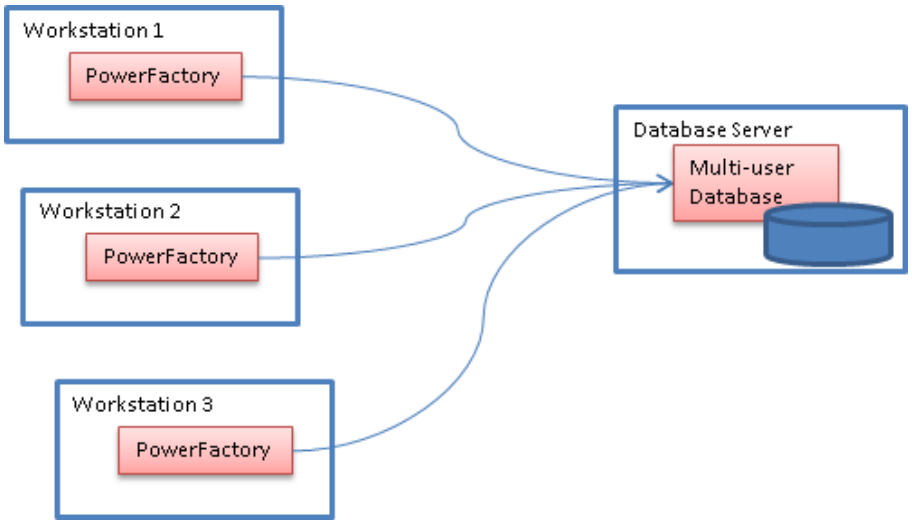

Figure 6.2.2: Multi-User Database

Advantages of a multi-user database are

- read-only or read-write sharing of projects
- project locking mechanisms
- better organisation of company-wide data (e.g. library, DPL scripts, template projects)
- backups have to be taken only from one database

PowerFactory supports two common commercial database systems:

- Oracle Database Server (see section 6.2.2 on page 22)
- Microsoft SQL Server (see section 6.2.3 on page 31)

6.2.1 Installation Overview

Generally, the following steps are required for setting up a multi-user environment:

1. Install *PowerFactory* (see section 8.3 on page 49). Verify that the application runs smoothly before proceeding with the next actions
2. Install and configure Oracle Database Server (see section 6.2.2 on page 22) or Microsoft SQL Server (see section 6.2.3 on page 31) and configure connection settings in *PowerFactory*
3. (Optionally) Set up and configure a shared *result files directory* or *archive directory* (see section 6.2.4 on page 36)
4. Administrate the *PowerFactory* database e.g. create *PowerFactory* user accounts and user groups.
5. (Optionally) Import projects, libraries, or other data from other *PowerFactory* installations

6.2.2 Oracle

6.2.2.1 Requirements

PowerFactory uses Oracle's OCCI/OCI programming interface to communicate with the server. OCCI/OCI itself uses a proprietary communication scheme on top of TCP/IP.

PowerFactory supports Oracle Client versions as shown below:

- Client: 19c
  - *PowerFactory* 64bit (x64) requires
    - Oracle Client for x64
  - *PowerFactory* 32bit (x86) requires
    - Oracle Client for x86
- Client: 18c
  - *PowerFactory* 64bit (x64) requires
    - Oracle Client for x64
    - Visual C++ Redistributable Package for Visual Studio 2013 x64
  - *PowerFactory* 32bit (x86) requires
    - Oracle Client for x86
    - Visual C++ Redistributable Package for Visual Studio 2013 x86
- Client: 12c Release 2:
  - *PowerFactory* 64bit (x64) requires
    - Oracle Client for x64
    - Visual C++ Redistributable Packages for Visual Studio 2013 x64
  - *PowerFactory* 32bit (x86) requires
    - Oracle Client for x86
    - Visual C++ Redistributable Packages for Visual Studio 2013 x86
- Client: 12c Release 1:
  - *PowerFactory* 64bit (x64) requires
    - Oracle Client for x64
    - Visual C++ Redistributable Packages for Visual Studio 2010 x64, and
    - Visual C++ Redistributable Packages for Visual Studio 2012 x64
  - *PowerFactory* 32bit (x86) requires
    - Oracle Client for x86
    - Visual C++ Redistributable Packages for Visual Studio 2010 x86, and
    - Visual C++ Redistributable Packages for Visual Studio 2012 x86

Hint: Oracle Client requires the Microsoft C++ Redistributable Packages mentioned above. These can be obtained directly from:
https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

---

PowerFactory supports Oracle Server versions as shown below:

- Oracle Client 19c supports Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
- Oracle Client 18c supports Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
- Oracle Client 12c Release 2 supports Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2
- Oracle Client 12c Release 1 supports Oracle Server 19c, 18c, 12c Release 2, 12c Release 1, 11g Release 2

Server Requirements: The Oracle server machine should fulfil requirements as follows:

- Dedicated server: the server machine should be used exclusively for *PowerFactory*.
- CPU: two or more cores/processors
- Memory: 2GB RAM or more

- Hard disk: 100 GB or more
- High network bandwidth (100 Mbit/s or higher) connection between the Oracle server and the machines where *PowerFactory* is running.

Depending on the number of users and projects the above numbers have to be adapted.

This section describes the installation and usage of the Oracle database server and client.

Before you set up and configure the client computers in section 6.2.2.3 (Client Installation), the server must be prepared as explained in section 6.2.2.2 (Server Installation).

6.2.2.2 Server Installation

Install the Oracle server software on the server computer (`ORACLESERVER` being the server computer name used in this section). Please follow the Oracle Server installation instructions.

Let us assume that `ORACLESERVER` is accessible by the name `oracleserver.domain.com` in the network. In order to use the Oracle server for *PowerFactory* you have to do the steps as follows:

- Create a new Oracle database with a database name (SID) of `PFSERVER` (stands for: *PowerFactory* Server) or use an existing one. The character set for the Oracle instance should be `WE8MSWIN1252`.
  - An Oracle listener is needed on the server, in order to pass on connection requests from clients to the database. If you created the database with the Oracle installer, the listener is configured for you automatically. If you create the database manually, then configure the listener using the Oracle Net Configuration Assistant tool.
  - (Recommended but not essential) create the database with redo log files sized at 500MB.
- *PowerFactory* needs a place to store its data. Oracle stores data in so-called tablespaces. It is suggested to create a new tablespace where *PowerFactory* (and only *PowerFactory*) stores its data. You might adapt and use the SQL statement below to your purposes:

```
CREATE TABLESPACE "POWERFACTORYTABLESPACE"
  LOGGING
  DATAFILE 'D:\ORACLE\ORADATA\DIGSI\POWERFACTORYTABLESPACE.DBF'
  SIZE 5000M REUSE
  EXTENT MANAGEMENT LOCAL
```

The statement creates a new tablespace named `POWERFACTORYTABLESPACE` which is stored in a `POWERFACTORYTABLESPACE.ora` file in the given directory. The file size is restricted to `5000M` i.e. about 5GB.

- ► In order to use the Oracle instance for *PowerFactory* one new Oracle schema is required. Create a new schema with the default profile. We suggest the schema name `PF`.
- ► Define a password for `PF`. In this example we use `aPasswordForPf`.
- ► Associate default and temporary tablespaces to schema `PF`. It is assumed that a temporary tablespace `TEMP` is available.
- ► Grant the roles `CONNECT` and `RESOURCE` and the system privileges `UNLIMITED TABLESPACE` and `ALTER SESSION` to `PF`. You might use and adapt the SQL script below to create the schema:

```
CREATE USER PF
  PROFILE DEFAULT
  IDENTIFIED BY aPasswordForPf
  DEFAULT TABLESPACE POWERFACTORYTABLESPACE
  TEMPORARY TABLESPACE TEMP
  ACCOUNT UNLOCK;
```

```
GRANT UNLIMITED TABLESPACE TO PF;
GRANT CONNECT TO PF;
GRANT RESOURCE TO PF;
GRANT ALTER SESSION TO PF;
```

- Start the Oracle instance process and Oracle listener process if they are not already started.

Note: The amount of tablespace space *PowerFactory* requires depends heavily on how *PowerFactory* is used. The space is roughly proportional to the number of objects in *PowerFactory*. Observe the tablespace fill-state regularly (e.g. once per month) and increase the size limit accordingly.

Note: Regarding the recommended sizing of the redo log files when creating the database; this configuration of the redo log files is to increase performance. *PowerFactory* can generate very large quantities of redo data in short bursts. This is particularly the case when importing data, copying large projects or deleting large projects. If for other reasons you require smaller redo log files, (for example to reduce database recovery time), you should increase the number of log file groups as an alternative to increasing the file size.

Note: It is highly recommended to configure security as described in section 6.2.2.6 on page 29.

Note: It is highly recommended to backup the Oracle schema on a daily basis. A backup procedure is described in section 6.2.2.7 on page 30

6.2.2.3 Client Installation

Two Oracle client packages can be used:

- (Normal) Oracle Client: This package includes many Oracle tools (e.g. management console, management tools, networking services, utilities etc.) which are not actually required for using *PowerFactory*; supports TNS names.
- Oracle Instant Client: This package contains only the files required for using *PowerFactory*; doesn't support TNS names.

Both (normal) Oracle Client and Oracle Instant Client are available for 32bit applications and 64bit applications. *PowerFactory* 64bit requires a 64bit Oracle Client; *PowerFactory* 32bit requires a 32bit Oracle Client.

Note: The required architecture of Oracle Client depends on *PowerFactory* only. This is not necessarily identical to the architecture of the Windows operating system. E.g. *PowerFactory* 32bit requires Oracle Client 32bit even if executed on Windows 64bit.

Install and configure Oracle Instant Client

Instant Client is a package of DLL files which can be downloaded freely from the Oracle website. The packages are ZIP archives that can be extracted anywhere e.g. to `c:\instantclient_12_1` (see figure 6.2.3). *PowerFactory* uses these DLL files in order to communicate with the Oracle Database server.

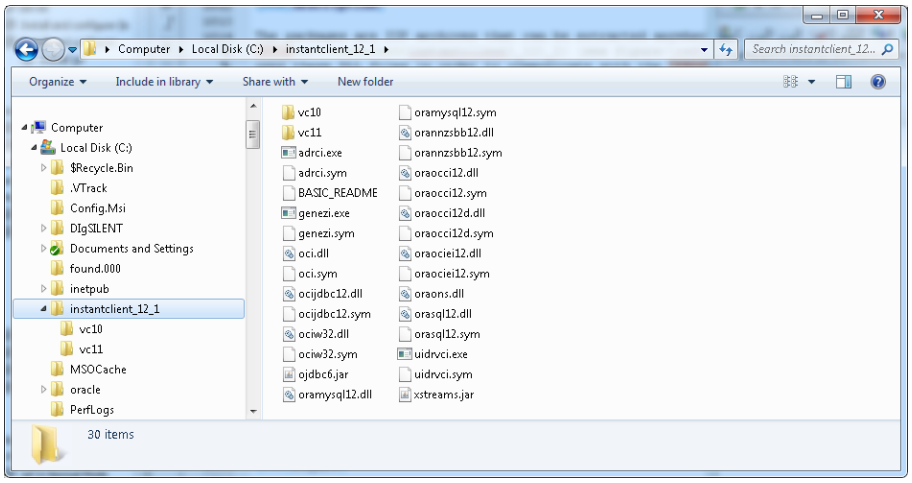

Figure 6.2.3: Database Figure: Oracle Instant Client installation directory

Install and configure (normal) Oracle Client

- Please use the newest version of the Client and follow the installation documentation. When asked for choose to install the "Runtime" installation option. It is assumed that the client software is installed in `C:\app\client\product\12.1.0\client_1` (see figure 6.2.4).

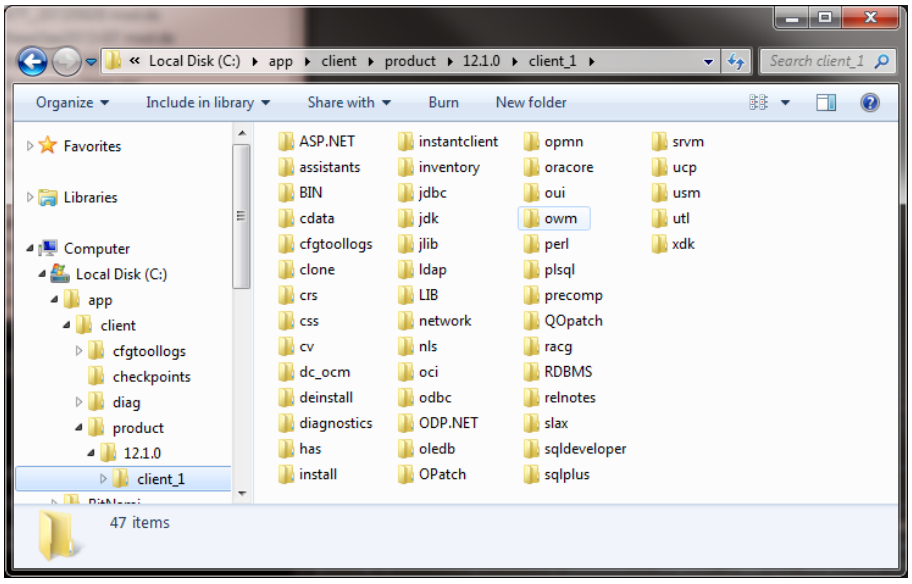

Figure 6.2.4: (Normal) Oracle Client installation directory

- (Optionally) It's possible to add an TNS name entry for `PFS`in the configuration file

```
C:\app\client\product\12.1.0\client_1\network\admin\TNSNAMES.ORA
```

The entry could be e.g.

```
PFS =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oracleserver)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = PFSERVER)
```

```
    )
  )
```

Then the `PFS` TNS name can be used in the *PowerFactory* configuration.

6.2.2.4 *PowerFactory* Configuration via Client Libraries

Start *PowerFactory* in configuration mode (see refsectionsec:conf).

- Switch to the **`Database`** page
- Insert the database connection settings as described below:

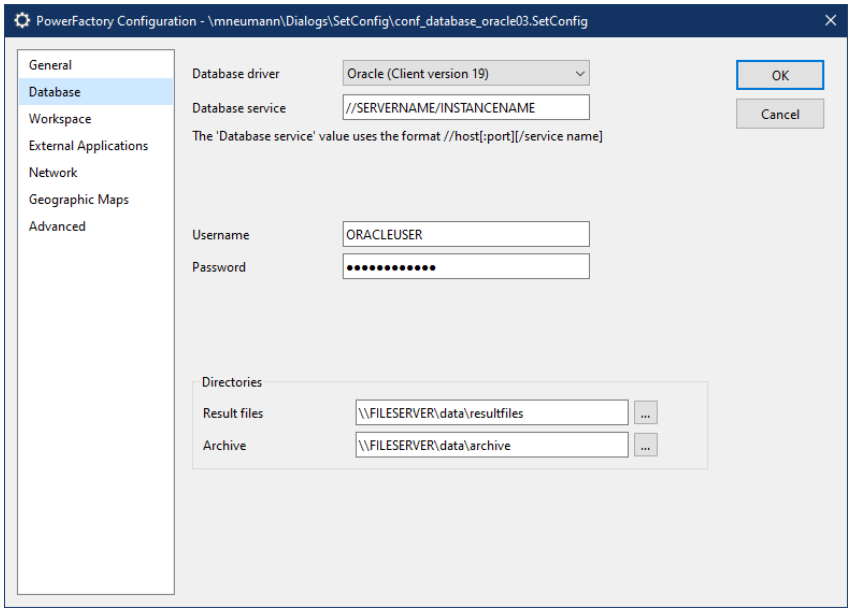

Figure 6.2.5: *PowerFactory* Configuration for Oracle Server

Database driver select `Oracle (Client Version 12c Release 1)` or `Oracle (Client Version 12c Release 2)`.

Database service this field describes the connection. It must be conform to the format

```
//host[:port][/servicename]
```

With the values used above (host=`oracleserver`, port=`1521` (default port), and SID=`PFSERVER`) the connection name is

```
//oracleserver/PFSERVER
```

If we had used a non-default port=`8888` the connection name would be

```
//oracleserver:8888/PFSERVER
```

If you've installed a (normal) Oracle Client and made an entry (e.g. `PFS`) in the `TNSNAMES.ORA` configuration file, you can use the TNS name instead. Then the **`Database service`** is just

```
PFS
```

Username and Password During the Oracle server setup an Oracle schema `PF` with the password `aPasswordForPf` has been created. Enter these values in the **`Username`** and **`Password`** fields.

PowerFactory uses files in the Oracle Client installation folder the communication with the server. In order to find them, the installation folders must be configured explicitly:

- Switch to the **`Advanced`** page
- Insert the folders as described below (see figure 6.2.6)

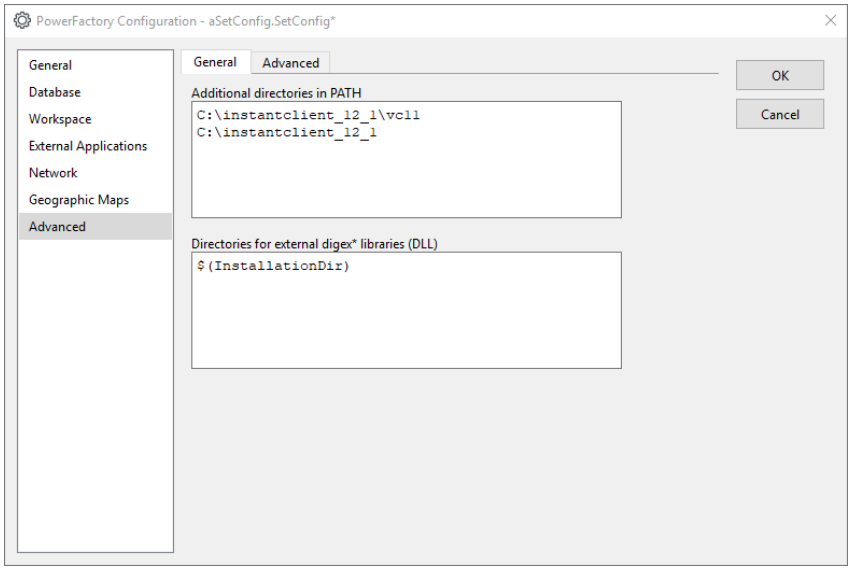

Figure 6.2.6: *PowerFactory* Configuration for Oracle Server

Additional directories in PATH

- ***Oracle Instant Client 19c*** installed e.g. in `C:\instantclient_19`:

```
C:\instantclient_19
```

- (Normal) ***Oracle Client 19c*** installed e.g. in `C:\app\client\product\19.0.0\client_1`:

```
C:\app\client\product\19.0.0\client_1\bin
```

- ***Oracle Instant Client 18c*** installed e.g. in `C:\instantclient_18`:

```
C:\instantclient_18
```

- (Normal) ***Oracle Client 18c*** installed e.g. in `C:\app\client\product\18.0.0\client_1`:

```
C:\app\client\product\18.0.0\client_1\bin
```

- ***Oracle Instant Client 12c Release 2*** installed e.g. in `C:\instantclient_12_2`:

```
C:\instantclient_12_2
```

- (Normal) ***Oracle Client 12c Release 2*** installed e.g. in `C:\app\client\product\12.1.0\client_1`:

```
C:\app\client\product\12.1.0\client_1\bin
```

- ***Oracle Instant Client 12c Release 1*** installed e.g. in `C:\instantclient_12_1`:

```
C:\instantclient_12_1\vc11
C:\instantclient_12_1
```

- (Normal) ***Oracle Client 12c Release 1*** installed e.g. in `C:\app\client\product\12.2.0\client_1`:

```
C:\app\client\product\12.1.0\client_1\oci\lib\msvc\vc11
C:\app\client\product\12.1.0\client_1\bin
```

6.2.2.5 *PowerFactory* Configuration via ODBC

Oracle also provides ODBC drivers for their database clients. The drivers and installation instructions can be downloaded from Oracle directly.

Note: The ODBC driver must be registered to use at least version 12c Release 1 of the Oracle client libraries.

The Microsoft Windows ODBC Data Source Administration tool (`odbcad32.exe`) provides a list of installed drivers. Once the installation of the Oracle ODBC driver has been completed they will be listed there. This tool can also be used to briefly test if the driver was installed correctly.

The ODBC driver name is the only additional information required in the PowerFactory database settings, everything else can be set up as described in 6.2.2.4. Make sure to spell the driver name exactly as shown in the ODBC Data Source Administration tool.

When using the Oracle ODBC driver, specifying the driver location in the Additional directories in PATH setting is not required.

6.2.2.6 Security

Network Encryption

The network traffic between *PowerFactory* and the database server is in plaintext and can be easily eavesdropped by any computer in the network, unless security is configured properly. Therefore it is strongly recommended to configure and enable Oracle's network security features.

Oracle provides data network *encryption* (Data is converted to ciphertext before it's sent over the network. It's highly infeasible that eavesdroppers are able to decrypt it.) and *integrity* (protection against data modification attacks and replay attacks) to ensure that data is secure as it travels across the network. Encryption and integrity can be easily enabled by adding some lines to the `listener.ora` configuration file on the server side:

```
# listener.ora (on the server!)

...

# Enforce encryption and integrity
SQLNET.ENCRYPTION_SERVER = required
SQLNET.ENCRYPTION_TYPES_SERVER = (AES256,AES128)
SQLNET.CRYPTO_CHECKSUM_SERVER = required
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER = (SHA256)
```

The above example enforces encryption with the AES256 or AES128 algorithm and integrity validation with the SHA256 algorithm. It's enforced for any connection to this database server.

Please consult the Oracle documentation for more details, especially the *Oracle® Database Security Guide* document. It also describes how to set up a certificate-based SSL/TLS authentication.

Transparent Data Encryption

Transparent Data Encryption (TDE) optionally encrypts data files on the hard disks of the database server. A encrypted tablespace can be create with the statement below:

```
CREATE TABLESPACE "POWERFACTORYTABLESPACE"
  LOGGING
  DATAFILE 'D:\ORACLE\ORADATA\DIGSI\POWERFACTORYTABLESPACE.DBF'
  SIZE 5000M REUSE
  EXTENT MANAGEMENT LOCAL
  ENCRYPTION USING 'AES256' ENCRYPT;
```

Besides it's also possible to encrypt an existing tablespace. Please consult the Oracle documentation for more details, especially the *Oracle® Database Advanced Security Guide* document.

6.2.2.7 Backup

Create Backup

Describing Oracle's backup facilities is far beyond the scope of this installation manual. In this section only a simple technique is described. Please consult the Oracle documentation for detailed information or other backup strategies. One backup method is the creation of database dumps. The `exp.exe` and `imp.exe` tools are part of the Oracle distribution. To export all data of the schema `PF` run the `exp.exe` tool:

```
exp.exe PF/aPasswordForPf@PFS file=d:\backups\database.dump owner=PF
```

where `PF` is the schema, `aPasswordForPf` is his password, `PFS` is the TNS name, and `d:\backup\database.dump` is the filename of the dump file.

Note: During the export or import process no *PowerFactory* user must be active. This backup strategy is suitable only for e.g. nightly backups.

Oracle also supports so-called hot backups where clients can still use the database during the backup process. Please consult the Oracle documentation for this backup strategy.

Restore Backup

To re-import a database dump, first drop and re-create the Oracle schema PF. Then run the `imp.exe` tool:

```
imp.exe PF/aPasswordForPf@PFS file=d:\backup\database.dump fromuser=PF touser=PF
```

If you import the dump into another Oracle instance ensure that there is a tablespace with the same name as the source instance.

6.2.3 Microsoft SQL Server

Microsoft provides several editions of its relational database system SQL Server. Depending on Version (2008 or newer) the availability of the editions may vary.

DIgSILENT *PowerFactory* is capable to use all editions as database engine. The free Express Edition provides almost the same functionality as the other editions but limits database size to 4 GB and lacks some of the more advanced administration tools. For a complete list of features for all editions consult the official SQL Server Homepage.

SQL Server operates as a service. Services are applications that run as background processes. The behaviour of services differs from that of other applications. For example, while most applications are executed only when a user launches the application from the Start menu, services such as SQL Server are generally started and stopped by the operating system environment. A service runs in the background and waits for processing requests. In the case of SQL Server, these requests are for database operations.

All actions described in this section are to be done on the server computer. Throughout this section *SERVERNAME* is used as computer name.

6.2.3.1 Server Installation

- ► Install the SQL Server software according to its documentation.

6.2.3.2 Server Configuration

Before the SQL Server service can be used it must be configured to allow for connections over the network.

- ► From the Start Menu select:
  - Microsoft SQL Server 20xx, depending on the version installed
  - Configuration Tools
  - SQL Server Configuration Manager
- ► Expand the SQL Server 20xx Network Configuration node
- ► Click on **`Protocols`** for **`SQLEXPRESS`** node (or the instance name you configured during installation respectively)
- ► On the right side, right click the **`TCP/IP`** entry and select **`Enable`** from the context menu
- ► Click on the SQL Server 20xx services node
- ► On the right side, right click the **`SQL Server (SQLEXPRESS)`** entry and select **`Restart`** from the context menu. The value in braces is the instance name, so make sure to select the correct one.
- ► On the right side, right click the **`SQL Server Browser`** entry and select **`Start`** from the context menu if it's not already running.

Your newly installed instance is now configured to allow network connections. With the next steps a database for *PowerFactory* is created.

- Again, from the **Start** Menu select:
  - Microsoft SQL Server 20xx
  - SQL Server Management Studio Express
- Change **Authentication** to SQL Server Authentication
- Enter the login name sa and enter the Password for sa, chosen during installation
- Select *File* → *New*→ *Query with Current Connection*
- Enter the following lines in the **Query Window** to the right:

```
sp_addlogin pf, aPasswordForPf
GO
CREATE DATABASE pfdb
GO
ALTER DATABASE pfdb SET ALLOW_SNAPSHOT_ISOLATION ON
GO
USE pfdb
GO
sp_grantdbaccess pf
GO
GRANT CREATE TABLE TO pf
GO
```

- Click **Execute** in the toolbar

You have now created a database named **pfdb** with a corresponding login named **pf** which uses the password **aPasswordForPf**. Feel free to change the values according to your needs.

Note: It is highly recommended to create daily backups of the database. The backup procedure for SQL Server is described in section 6.2.3.5

6.2.3.3 *PowerFactory* Configuration

Start *PowerFactory* in configuration mode (see section 8.6 on page 62).

- Switch to the **Database** page
- Insert the database connection settings as described below (see figure 6.2.7)

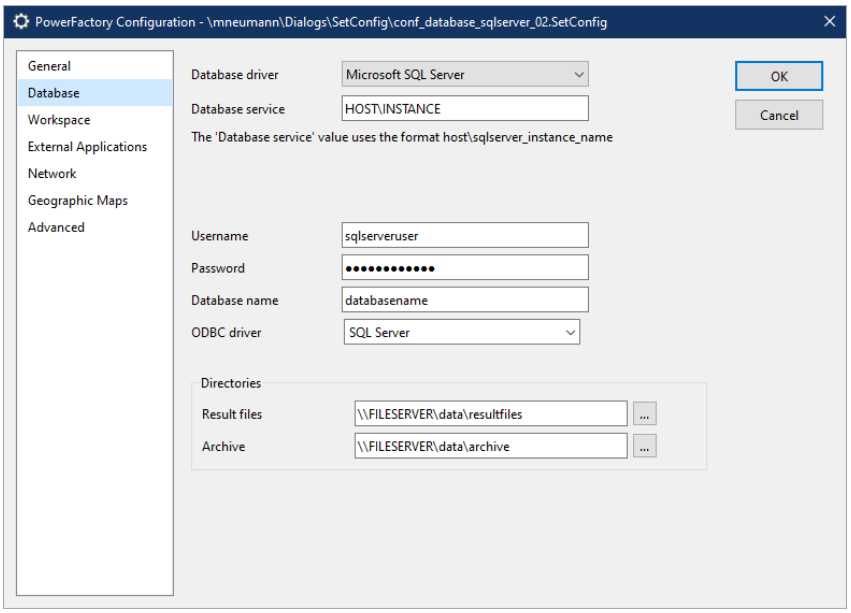

Figure 6.2.7: *PowerFactory* Configuration for SQL Server

Database driver select `Microsoft SQL Server`

Database service The **`Database service`** uses the format `server_name\instance_name`. If you used the default values SQL Server provides, it would be `SERVERNAME\SQLEXPRESS` for the 2008 edition. SERVERNAME is our virtual server name during this manual, as stated earlier.

Username, Password, and Database name for this manual it would be `pf` as username, `aPasswordForPf` as password and `pfdb` as database name.

6.2.3.4 Security

Network Encryption

SQL Server supports encryption of connections using the TLS protocol. The Microsoft support pages list in detail which versions of SQL Server support which TLS versions.

Encryption can be enabled on the server side and requires a trusted Certificate Authority signed certificate for maximum security. SQL Server allows the usage of self-signed certificates but this option is not recommended for production environments as it is prone to man-in-the-middle attacks.

Please note that PowerFactory cannot enforce a connection to be encrypted. It's the server that specifies if encryption is required or not. This way encryption can easily be enabled at any time, and also be removed if there is no further need for it.

Transparent Data Encryption

Transparent Data Encryption (TDE) optionally encrypts data files on the hard disks of the database server. A database can be encrypted with the statement below:

```
ALTER DATABASE pfdb SET ENCRYPTION ON;
```

Please consult the SQL Server documentation for more details.

6.2.3.5 Backup

Create a Backup

As with Oracle, **SQL Server's** backup facilities are far beyond the scope of this documentation. It is highly suggested you consult the official Microsoft documentation for detailed backup strategies. We present a very simple method to backup a database and must stress that this method only satisfies the most basic needs.

A full SQL Server database backup can be easily created from a DOS console on the server computer.

- Open a command prompt and enter (all in one line):

```
sqlcmd -S SERVERNAME\SQLEXPRESS -U sa -P aPasswordForSa -e
  -Q "BACKUP DATABASE pfdb TO DISK='d:\backupdir\pfdb.dump' WITH INIT"
```

This dumps the database named **pfdb** to a file `d:\backupdir\pfdb.dump`, **sa** and **aSecurePasswordForSa** are username and password of the database administrator user.

It is suggested to create a batch script which performs this task. To do so:

- Open a new batch file e.g. `d:\backupdir\backup.bat` with a text editor (e.g. Windows' **Notepad** editor).
- Insert the above text into the file (all in one line!).
- Close the file.

Windows' **Scheduled Tasks** utility allows you to run this batch script automatically, for example every day at midnight.

- Open the scheduler manager Windows Start menu:
  Start → *Programs*→ *AccessoriesSystem Tools* → *Scheduled Tasks*
- Add a new scheduler task *PowerFactory* backup and configure it as follows (see the Windows documentation for further information)
- Enter the backup script `d:\backupdir\backup.bat` to be run.
- Select the **daily** option and **00:00** as start time.

This configuration creates nightly backups. The `pfdb.dump` file is overwritten each time. If you want to keep the latest three backup dumps enhance the above `backup.bat` script as follows.

- Open the batch script created earlier
- Replace its contents with the following lines (again, the sqlcmd command in one line)

```
copy d:/backupdir/pfdb2.dump d:/backupdir/pfdb3.dump
copy d:/backupdir/pfdb1.dump d:/backupdir/pfdb2.dump
copy d:/backupdir/pfdb.dump d:/backupdir/pfdb1.dump
sqlcmd -S SERVERNAME\SQLEXPRESS -U sa -P aPasswordForSa -e
  -Q "BACKUP DATABASE pfdb TO DISK='d:/backupdir/pfdb.dump' WITH INIT"
```

- Save the file.

Restore a Backup

The **RESTORE_DATABASE** command recovers a database from a dump file. To restore a database backup on the same database server where it was created follow the instructions below.

- Run `sqlcmd` from the command prompt

```
sqlcmd -S SERVERNAME\SQLEXPRESS -U sa -P aPasswordForSa
```

- Run the following commands inside `sqlcmd`

```
DROP DATABASE pfdb
GO
RESTORE DATABASE pfdb FROM DISK='c:/backupdir/pfdb.dump' WITH RECOVERY
GO
```

Restoring the database on a different SQL Server installation is more complicated. Let `c:\msde2` be the installation folder of the target SQL Server. All commands are to be run inside `sqlcmd`.

- First create a database user **pf** for *PowerFactory* usage.

```
sp_addlogin pf, aPasswordForPf
GO
```

- Import the database dump as follows:

```
RESTORE DATABASE pfdb
  FROM DISK = 'd:\backup\pfdb.bak'
  WITH MOVE 'pfdb' TO 'C:\msde2\mssql\data\pfdb.mdf',
  MOVE 'pfdb_log' TO 'C:\msde2\mssql\data\pfdb.ldf',
  RECOVERY
GO
```

- Adjust the access rights of the **pf** user

```
USE pfdb
GO
sp_change_users_login AUTO_FIX, pf
GO
```

6.2.4 Directories for Result Files and/or Archive (optional)

A *result files directory* is a shared directory where all *PowerFactory* instances can read and write files.

PowerFactory stores almost all data in the database. *Result Files* are an exception. *Result Files* are binary files containing the result of simulation calculation. Due to performance reasons (these results can be arbitrarily large) they are not stored in the database but directly in files on the hard disk of the local computer.

Result Files are actually redundant since they are re-created when re-running the calculation. However since a simulation calculation can take hours to finish, it's worth to keep them at hand.

In a multi-user database scenario where two *PowerFactory* users USER1 and USER2 work on the very same project, USER2 cannot access the Result Files created by USER1 because they're stored on USER1's hard disk.

Result files in a *result files directory* can be accessed by all *PowerFactory* users working on the same multi-user database (see figure 6.2.8).

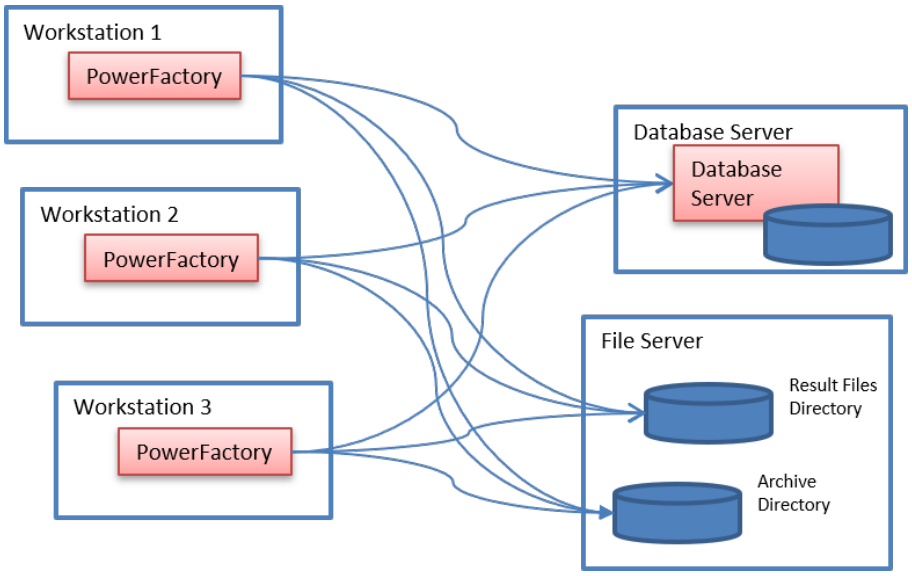

Figure 6.2.8: Result files and archive directory

Similarly there's a directory for Project Archiving: users can "archive" their projects. The projects are exported into the *archive directory* and then deleted from the database. Users can restore them later. The Housekeeping configuration allows to archive projects that have not been used for a long time automatically.

Any shared directory (e.g. on a file server) can be used as *result files directory* or *archive directory* if all *PowerFactory* users (i.e. the Windows users running *PowerFactory*) have read and write access to it. The directory paths have to be configured in *PowerFactory*.

Start *PowerFactory* in configuration mode (see section 8.6 on page 62).

- Switch to the **Database** page
- Insert the database connection settings as described below (see figure 6.2.8)

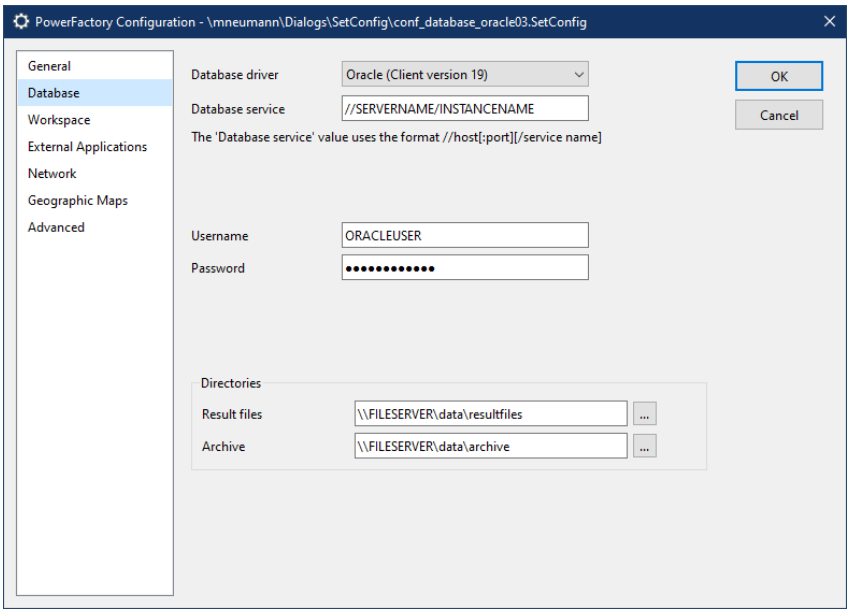

Figure 6.2.9: *PowerFactory* Configuration for SQL Server

Directories: Result files specify a directory path e.g. a mapped network drive

```
E:\data\resultfiles
```

or the raw network path.

```
\\fileserver\data\resultfiles
```

The specified directory must exist and the current Windows user must have read and write access to this directory.

Directories: Archives specify a directory path accordingly.

Note: A *result files directory* is required only in rare scenarios e.g. when *PowerFactory* users often run long-running simulations and work on the very same shared project. In all other cases this directory is not needed.

Note: An *archive directory* is only required if using the project archiving feature. In all other cases this directory is not needed.

Note: It's strongly suggested to create backups of the directories on a regular basis.

6.3 Read-only Mode

PowerFactory can be started in a "read-only mode". In this mode the application reads data from an existing database, but never writes data back to the database i.e. all data changes are lost when the application is closed.

Its purpose is a parallel calculation on the same data e.g. several *PowerFactory* engines perform similar calculations on the very same project. Without read-only mode this scenario would require for each concurrent engine a separate *PowerFactory* user, each owning a separate copy of the project. With read-only mode all engines can use the same project and the same *PowerFactory* user.

Note: In read-only mode the functionality is restricted e.g. database-related operations like creation of versions, PFD export etc. are not available.

Though all engine instances read from the very same database, each engine instance requires a separate instance directory within the workspace directory where instance-specific temporary data is stored e.g. the log file or other temporary files.

Usage: the read-only mode is enabled by the command line switch `/readonlymode`. The instance identifier can be set with `/instance`. Please note that the instance argument is mandatory for starting multiple instances from same working directory. Example:

```
PowerFactory.exe /readonlymode /instance INST1
```

PowerFactory uses now a `db-INST1` directory. If it's not there it's created automatically:

```
WORKSPACE
+- db                  // normal workspace data
|  +- log
|  +- tmp
|  +- ...
|
+- db-INST1            // workspace data for instance INST1
   +- log
   +- tmp
   +- ...
```

C++-API example: start a read-only engine instance for user `USERNAME` and password `PASSWD` with the instance identifier `INST1`.

```
CreateApiInstanceV1(
  "USERNAME",
  "PASSWD",
  "/readonlymode /instance INST1");
```

Chapter 7

Security and Privacy

7.1 Audit Log

PowerFactory is capable to record security-relevant events in an Audit Log (e.g. failed user authentication or data modifications in the global `Configuration` folder). The Audit Log is stored in the database in a secure way, protected against accidental or malevolent manipulation.

7.1.1 Configuration

The Audit Log function is configured in the `SetAuditlog` object located in the configuration section of the database:

```
Database
+- Configuration
   +- Security & Privacy
      +- Audit Log configuration.SetAuditlog
```

This object is also conveniently available from *PowerFactory*'s main menu via *ADMINISTRATION* → *Security & Privacy*→ *Auditlog*:

- The Audit Log can be enabled or disabled.
- If enabled, a retention period between 1 and 365 days can be specified. Events older than that period are purged regularly. Records are purged when the Administrator user logs on or during the Housekeeping job (see section 8.2.0.1 on page 47).

7.1.2 Events

The Audit Log stores information about a fixed set of security-related events that might occur during a *PowerFactory* session. For each event several attributes are recorded as shown below:

- `Time`: A UTC timestamp.
- `SessionId`: A unique ID that identifies a session unambiguously. It's required to correlate events, especially when many users work concurrently.
- `SequenceId`: Within a session each event is numbered.
- `Category`: An event category e.g. `authentication.successful` or `user.addToGroup`.

- `Details`: Category-dependent additional information.

The remainder of this section describes all categories in detail.

`application.*`: Application startup and shutdown

- `application.start`: start of *PowerFactory*. It also could be C++-API application or a standalone Python script. The following details are recorded:
  - `windowsUsername`: name of the Windows user.
  - `computerName`: name of the Computer.
  - `executable` name and path of executable e.g. `C:\ProgramFiles\DIgSILENT\PowerFactory2019SP2\PowerFactory.exe`.
- `application.stop`: application is closed. If the application stops unexpectedly e.g. due to a crash or loss of network connection to the database server, there'll be no such event.

`authentication.*`: Authentication-related events:

- `authentication.successful` successful authentication of a user.
  - `username`: *PowerFactory* user name
- `authentication.unsuccessful.unknownUsername`: the specified user can not be found in the database. (The user name itself is not recorded: it might be the password, if the user confuses the user name field with the password field, and the clear text password would be recorded in the Audit Log.)
- `authentication.unsuccessful.wrongPassword`: wrong password for user name
  - `username`: user name, but only if it is a known user in the database. (It might be the password, if the user confuses the user name field with the password field, and the clear text password would be recorded in the Audit Log.)
- `authentication.unsuccessful.userDisabled`: authentication failed because user is disabled.
  - `username`
- `authentication.unsuccessful.userExpired`: authentication failed because the user is enabled only for a restricted time period.
  - `username`
- `authentication.unsuccessful.refusedToSetNewPassword`: authentication was successful, but the user refused to change the password according to the configured password policy.
  - `username`
- `authentication.userSharing`: User successfully authenticated as user X but works as user Y.
  - `authenticatedUser`
  - `effectiveUser`

`session.*`: Categories for session events

- `session.terminate`: Administrator explicitly kills other user's session.
  - `username`
- `session.terminateFormerSession`: A user is already logged on and starts a second session, terminating the former session.
- `session.terminated`: This session was killed by one of the above events.
- `session.idleSessionTimeout`: Application was closed because it was idle for too long.

`user.*`: Changes in user configuration:

- `user.create`: Administrator creates user
  - `username`
- `user.modifyRelevantAttribute`: security-relevant modification of user XYZ (IntUser object). Security-relevant attributes:
  - `username`
  - `attribute` attribute name
  - `previousValue` previous value
  - `newValue` new value
- `user.moveToRecycleBin`: Adminstrator deletes user object.
  - `username`
- `user.restoreFromRecycleBin`: Administrator restores user object from recycle bin
  - `username`
- `user.deleteFromRecycleBin`: Administrator empties recycle bin containing a user object.
  - `username`
- `user.addToGroup`: Administrator adds user to group.
  - `username`
  - `groupname`
- `user.removeFromGroup`: Administrator removes user from group.
  - `username`
  - `groupname`

`configuration.*`: Changes in configuration (`Database\Configuration`)

- `configuration.create`: Add object in configuration.
  - `object`: full object path
- `configuration.modify`: Modify object in configuration.
  - `object`: full object path
- `configuration.moveToRecycleBin` Delete object in configuration and move it to the recycle bin.
  - `object`: full path to the object before deletion
- `configuration.restoreFromRecycleBin`: Restore configuration object from recycle bin.
  - `object`: full path to the object after restore.

`auditlog.*`: Audit Log functionality

- `auditlog.report`: Audit report occurred.
- `auditlog.export`: Audit export occurred.
  - `minTime`, `maxTime`: time filter in UTC time.
  - `filepath` the path of the export file.
  - `size`: the size of the export file.
  - `sha256Checksum`: SHA256 checksum of exported file.
- `auditlog.automaticPurge`: Automatic Audit purge occurred
  - `retention`: retention time in days.
  - `time`: sessions starting before this time (in UTC) were purged.
  - `numberOfDeletedSessions`: number of sessions that were purged.

`duk.*`: Database Unlock Key (DUK) functions

- `duk.unlock`: The Administrator password was reset using the DUK.
- `duk.reset`: The DUK was changed.

Other functionality:

- `data.export` User exports data to PFD or DZ file.
  - `objects`: full path of top-level objects that are exported
  - `filepath`: export file name

7.1.3 Report, Export, and Integrity Check

Several functions are provided for evaluating the Audit Log.

The `ComAuditlog` command object provides functions for retrieving information from the Audit Log. These functions are available only for the Administrator user.

Report Writes some statistics about the Audit Log to the Output Window e.g. date of first and last recorded event or count of each event category.

Export Writes the content of Audit Log to a text file which can be imported e.g. in Microsoft Excel for further analysis. Regular text exports can be archived for a later reference.

Integrity Check *PowerFactory* stores Audit Log events on the database server. It uses encryption and checksum in order to ensure data integrity. With this check function it's possible to detect data manipulation e.g. modification or deletion of events.

7.2 Idle Session Timeout

This security function automatically closes a running *PowerFactory* application when the user does not interact with it by mouse or keyboard for a configurable time period. The application is not closed during a long-running operation e.g. a complicated calculation or a DPL/Python script run.

The Administrator can configure it globally for all *PowerFactory* users with the `SetSessionidle` object located in the configuration section of the database:

```
Database
+- Configuration
   +- Security & Privacy
      +- Session Idle Configuration.SetSessionidle
```

This object is also conveniently available from *PowerFactory*'s main menu via *ADMINISTRATION* → *Security & Privacy*→ *Idle Session Timeout*.

It's possible to override this global setting for each user separately in the `IntUser` object.

7.3 Privacy

The Administrator can configure privacy globally for all *PowerFactory* users with the `SetPrivacy` object located in the configuration section of the database:

```
Database
+- Configuration
   +- Security & Privacy
      +- Privacy Configuration.SetPrivacy
```

This object is also conveniently available from *PowerFactory*'s main menu via *ADMINISTRATION* → *Security & Privacy*→ *Privacy*.

Enable recording of modifying user in object If enabled: persistently stores the *PowerFactory* username of the user who modified the object last.

Display system account in User object If enabled and the user has started a *PowerFactory* session: the system account (the Windows username) of the user is shown in the `IntUser` object.

7.4 Login Policy

PowerFactory 2021 supports three different login modes, which can be configured in the `SetLoginpolicy` object located in the configuration section of the database:

```
Database
+- Configuration
   +- Security & Privacy
      +- Login Policy.SetLoginpolicy
```

This object is also conveniently available from *PowerFactory*'s main menu via *ADMINISTRATION* → *Security & Privacy*→ *Login Policy*. When initializing a multi-user database it will automatically be shown up in the course of database initialisation.

The default setting equals the behavior of former *PowerFactory* versions.

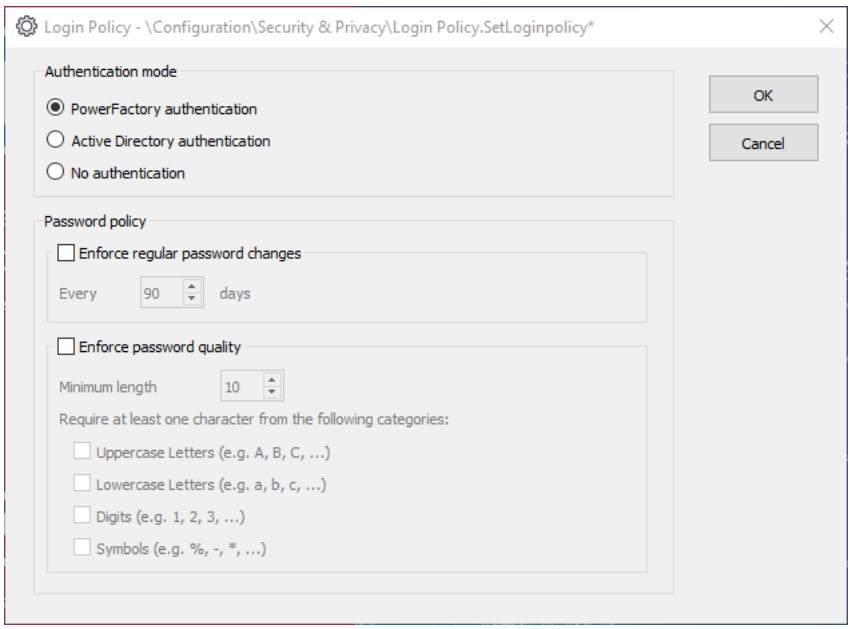

Figure 7.4.1: Selection dialogue for login policy to *PowerFactory* (default settings)

7.4.1 PowerFactory authentication

This mode uses the built-in authentication mechanism of *PowerFactory*. The *PowerFactory* administrator can configure the password policy according to the company security guide lines.

In case there are users with higher requirements regarding password quality these settings can also be overwritten in the `IntUser` object (subpage "Password Policy") for individual users. The same applies for the enforcement to set a new password. The administrator can force an individual user to set a new password with the next login by setting the flag "Force password change" in the "General" page of the `IntUser` object.

Hint: The default setting does not enforce password quality or length. To meet common security requirements a password should have at least a length of 10 characters and enforce characters from at least three of the offered categories. Regular password changes (after 90 days) are also highly recommended.

Password recovery

Forgotten passwords can easily be reset by the *PowerFactory* administrator.

For the case that the *PowerFactory* administrator password gets lost, *PowerFactory* offers the possibility to set a so-called database unlock key (DUK) on database initialisation. It's a 32-character string with high entropy that is meant to be stored in a secure place until needed.

When initializing a multi-user database, the initial setup dialogue is shown in *PowerFactory*(see 7.4.2. Here the *PowerFactory* administrator will be forced to set the administrator password as well as a database unlock key (use the 'Generate' button to generate a random key).

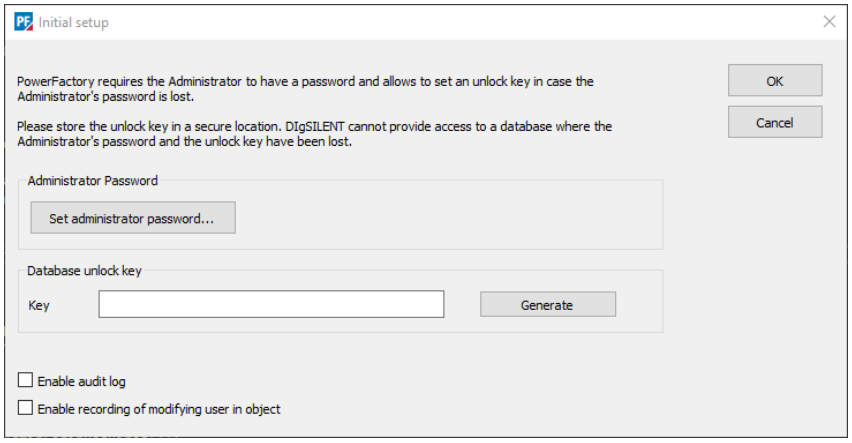

Figure 7.4.2: Initial Setup Dialogue

On clicking 'OK' the specified DUK will automatically be copied to the clipboard so it can be pasted into a text document using Ctrl+V.

To unlock a database using the DUK or to reset the DUK use the command line options as specified in section 8.7.13 on page 74.

Hint: *DIgSILENT* CANNOT provide access to a database when the administrator's password and the unlock key have been lost!

7.4.2 Active Directory authentication

This authentication mode is suitable for customers who are already using Microsoft Active Directory and would like to manage *PowerFactory* users in this same central system.

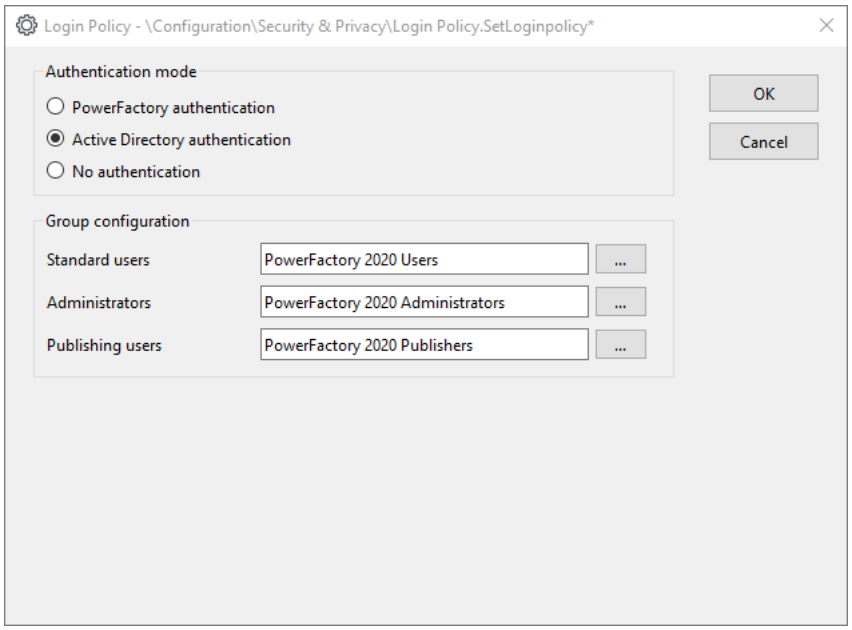

Figure 7.4.3: Selection dialogue for login policy to *PowerFactory* (Active Directory authentication settings)

PowerFactory supports to configure certain Active Directory user groups, so members of these groups can run *PowerFactory* and log on to the application without providing username and password.

The configuration allows to specify user groups for

- Standard Users: Users of this group are granted log-in rights to PowerFactory using their domain username.
- Administrators: Users of this group are allowed to log-in to PowerFactory as user Administrator. There can only be one user logged in as Administrator at any given time.
- Publishing Users: Users of this group are allowed to log-in as any publishing user.

PowerFactory will always check certain Active Directory groups so it is always possible to restore access to the application should the configuration be defective. For PowerFactory 2020 these user groups are named

- PowerFactory 2020 Users
- PowerFactory 2020 Administrators
- PowerFactory 2020 Publishers

Hint: Active Directory authentication has been introduced in PowerFactory 2019. For all versions after PowerFactory 2019 the default groups are named according to the major version.

7.4.3 No authentication

This mode is meant for users that explicitly want to avoid authentication for *PowerFactory* (is default for single-user, local database usage).

Chapter 8

Advanced Options

Most installation variants in this chapter usually make only sense in a multi-user context with *PowerFactory* running in a server environment.

- Instead of activating a *workstation licence* on each computer with a *PowerFactory* installation, you can use a central licence server that provides a *network licence* for all users in your network (see section 4 on page 7).
- *PowerFactory* stores user data in a Workspace on the hard disk. The Workspace functionality is described in section 6.1.1 on page 19.
- A *multi-user database* allows several *PowerFactory* users to work concurrently and share their data (see section 6.2).
- Several users can work on the same *application server* (see section 8.3 on page 49).
- *Offline Mode* is based on a *multi-user database*. It allows to run *PowerFactory* without a permanent connection to the server (see section 8.4 on page 51).

8.1 Application Language

The *PowerFactory* application does not support Unicode. Running *PowerFactory* with e.g. Chinese as application language requires that Windows' "Language for non-Unicode Programs" is configured accordingly.

- Chinese: requires "Chinese (Simplified, China)"
- French: requires "French (France)"
- German: requires "German (Germany)"
- Russian: requires "Russian (Russia)"
- Spanish: requires "Spanish (Spain)"
- Turkish: requires "Turkish (Turkey)"

English as application language needs no special configuration. The "Language for non-Unicode Programs" can be configured as follows (see figure 8.1.1):

1. Open Windows' "Region & language settings".
2. Follow the "Administrative language settings" link. A small "Region" dialogue is shown.
3. Select the "Administrative" page.
4. Select the "Language for non-Unicode programs" as described above. This might require a Windows restart.

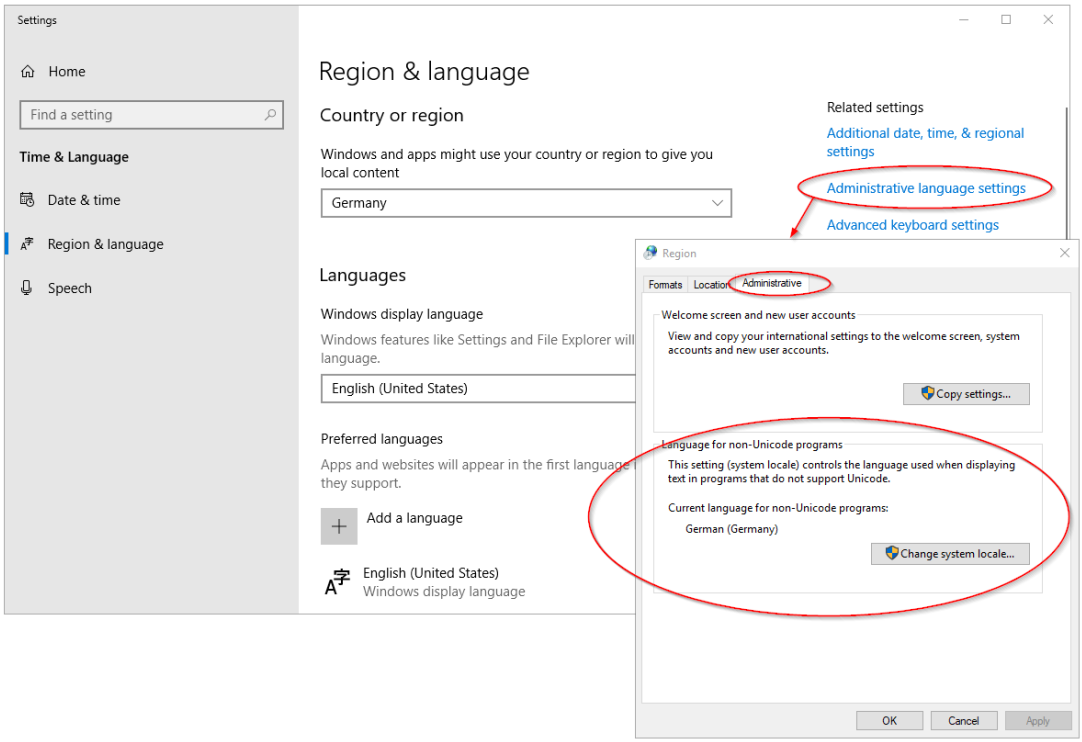

Figure 8.1.1: Language for non-Unicode programs

8.2 Housekeeping (optional)

Over the course of time the database grows and is cluttered with old data that might have a general negative impact on the database performance. *PowerFactory* provides a Housekeeping job that can be configured to be run as periodical background job e.g. every night.

- Deletion of items older than a configurable age in all users' recycle bin.
- Archiving (or even complete deletion) of projects that have not been activated recently.
- Regular purge of project storage.

Housekeeping has to be configured in *PowerFactory* database by the Administrator as described in the PowerFactory *User Manual*.

8.2.0.1 Scheduling Housekeeping

This is an optional installation step that can be carried out at a later date. Housekeeping is described in the Program Administration chapter of the PowerFactory *User Manual*.

Housekeeping is executed via a Windows Scheduled Task from a computer with *PowerFactory* installed. Typically this will be a terminal server (e.g. Citrix) or some other application server. Housekeeping makes use of a command line initiation of *PowerFactory*. An example execution is as follows:

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe" /housekeeping:8:60
```

In the example above, `8` specifies the maximum run duration of the housekeeping as eight hours. If the housekeeping is scheduled to start at 11 p.m. and is still not finished at 7 a.m., it will exit after

completing its current action.

In the example above, `60` specifies the sleep period, in seconds, after a housekeeping action. If there were no sleep period the housekeeping would place a heavy workload on the system, possibly affecting other active users.

Housekeeping connects as a special administrative *PowerFactory* user called *Housekeeper*, which is automatically created when housekeeping is run for the first time.

The housekeeping execution should be triggered via a Windows Scheduled Task (Windows 2008: Control Panel/Administrative Tools/Task Scheduler/Create Task). An example of the action configuration is shown in figure 8.2.1.

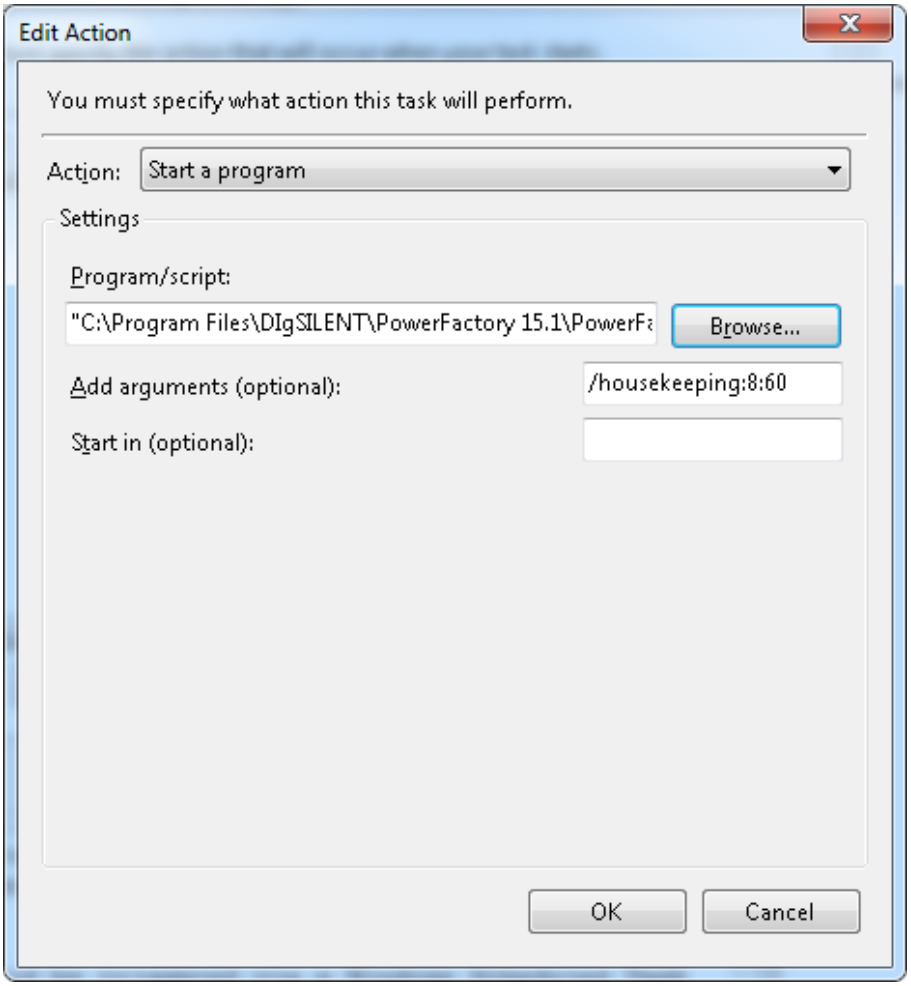

Figure 8.2.1: Configuring a scheduled task to run housekeeping

Program/script Insert the path to the *PowerFactory* executable e.g.

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
```

The leading and trailing quote characters (`"`) are required since the path contains space characters.

Add arguments (optional) Insert the housekeeping parameters e.g.

```
/housekeeping:8:60
```

Windows scheduled tasks can be automatically stopped. It is preferable to configure this via the housekeeping maximum run duration parameter, but the scheduled task configuration can be used as a 'backstop'. When configuring the scheduled task, choose an appropriate operating system user to run the task. The user does not need to be logged on.

8.3 Application Server

An *Application Server* (e.g. Citrix Application Server) allows to run an application several times in parallel Windows sessions. A typical scenario for a *PowerFactory* installation is shown in figure 8.3.1. Several machines and components are involved:

Many Workstations The actual *PowerFactory* user works locally on his Workstation. When the user starts the application, a *PowerFactory* process is started on the Application Server. The application windows is shown on the Workstation.

One or more Application Servers One or more instances of *PowerFactory* are running on this server. There might be several Application Server Computers organised in a Server Farm.

Database Server Manages the central *PowerFactory* database. All *PowerFactory* instances communicate with it.

File Server (Optional) provides a shared *result files* or *archives directory* (section 6.2.4 on page 36).

Licence Server Installed along with the Licence Components (see section 4 on page 7). The ***Licence Server*** doesn't have to run necessarily on a separate computer, it can be hosted on the Application Server.

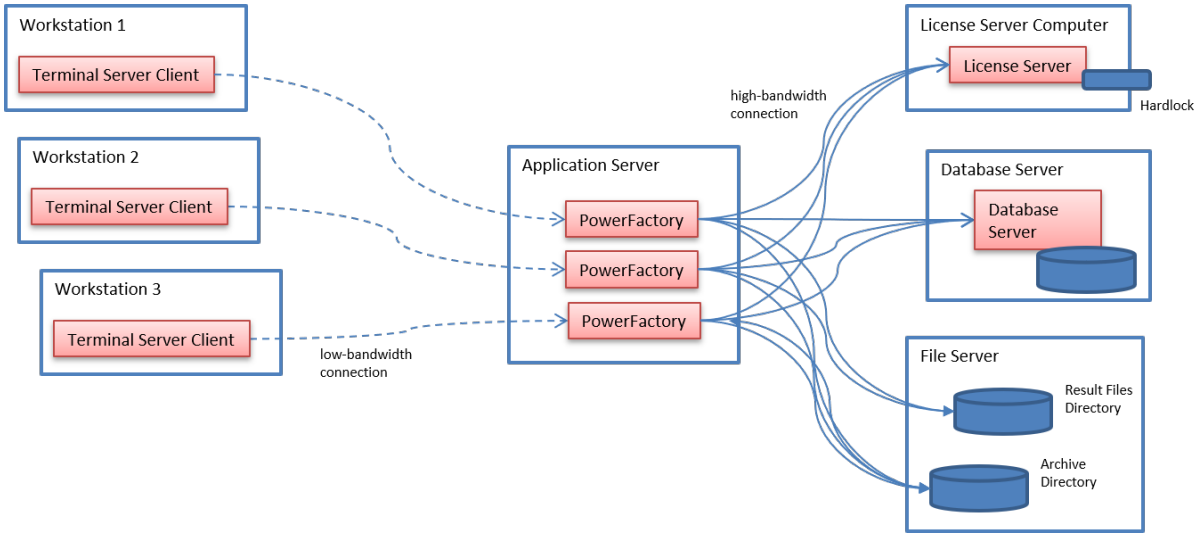

Figure 8.3.1: Application Server Environment

Installing *PowerFactory* on an Application Server offers several advantages over a "normal" multi-user database installation:

- *PowerFactory* has to be configured only once on the Application Server computer, but can be used by potentially hundreds of Workstations.
- A high-bandwidth network connection is necessarily required between *PowerFactory* and a database server. Bigger companies have their *PowerFactory* users distributed over several remote locations with low-bandwidth network.

Figure 8.3.1 suggests that all components have to be installed on different machines. But it's possible to deploy several components on the same computer e.g. the Application Server machine can host *PowerFactory*, ***Licence Server***, and the Vault directory.

Note: *PowerFactory* is executed in its entirety on the application server. It is important that the server complies with *PowerFactory*'s computing requirements: RAM, CPU(s), hard disk space, etc.

This section describes how to configure *PowerFactory* on application server for three Windows users *Frodo*, *Sam*, and *Pippin*. It is assumed that

- *PowerFactory* is already installed on the computer.
- Additionally a multi-user database should be up and running as described and *PowerFactory* is configured to use it section 6.2 on page 21.

The *PowerFactory* installation directory (usually `C:\Program Files\DIgSILENT\PowerFactory 2019`) contains

- *PowerFactory* the application binary `PowerFactory.exe` along with several `DLL` files
- the configuration file `PowerFactory.ini`
- other data e.g. a template for initial database content, Demo examples etc.

```
C:\
+ Program Files
   +- DIgSILENT
      +- PowerFactory 2019
         +- PowerFactory.exe    // executable
         +- PowerFactory.ini    // configuration
         +- ...
```

Each Windows user that runs *PowerFactory* requires a separate workspace directory which will hold temporary data and log files. The workspace directories are subfolder of a common workspace directory e.g. `d:\Data\PowerFactory Workspaces`:

```
D:\
+- Data
   +- PowerFactory Workspaces
      +- Frodo                  // Frodo's workspace directory
      +- Sam                    // Sam's workspace directory
      +- Pippin                 // Pippin's workspace directory
      +- ...
      +- resultfiles            // (optional) common result files directory
      +- archive                // (optional) common archive directory
```

The `resultfiles` and `archive` directories are optional (see section 6.2.4 on page 36). The workspaces directory must not be necessarily on a local hard disk. It can be on an network drive instead e.g. `\\SERVER\PowerFactory Workspaces`.

8.3.1 Workspace directories

► Create a root directory for all Workspaces e.g.

```
D:\Data\PowerFactory Workspaces
```

- Create a Workspace directory for each Windows user:

```
D:\Data\PowerFactory Workspaces\Frodo
D:\Data\PowerFactory Workspaces\Sam
D:\Data\PowerFactory Workspaces\Pippin
```

Verify that each Windows user has read and write access to his Workspace directory

- (optional) Create a *result files* directory e.g.

```
D:\Data\PowerFactory Workspaces\resultfiles
```

Verify that each Windows user has read and write access to the Vault directory.

8.3.2 *PowerFactory* Configuration

Start *PowerFactory* in configuration mode (see section 8.6 on page 62).

- Switch to the **Workspace** page
- Uncheck *Use Default Workspace Directory*
- Insert as *Directory*

```
D:\Data\PowerFactory Workspaces\%USERNAME%
```

`%USERNAME%` will be replaced by the current Windows username.

`%USERNAME%` will be replaced by the current Windows username.

(Optional) If your using a *result files* or *archive directory*, set the directories on the **Database** page as described in section 6.2.4 on page 36.

8.4 Offline Mode

Since Version 15.0 *PowerFactory* provides the ability to work in *Offline Mode* when a network connection to the database server is unavailable. The required project data is cached to the user's local machine, which can then later be synchronised to the server database. *Floating Licences* can be generated which allow to work without a permanent connection to the ***Licence Server***. More information on *Floating Licences* can be found in chapter 5.7.

Note: Offline Mode requires the *Multi-user database* module and the *Floating Licence Server* feature.

Note: Offline Mode can not be combined with the *Project Archiving* functionality.

This section describes the installation and configuration of the ***Offline Proxy Service***, a software component of *PowerFactory* to be used with the Offline Mode database driver. Figure 8.4.1 gives an overview over all components.

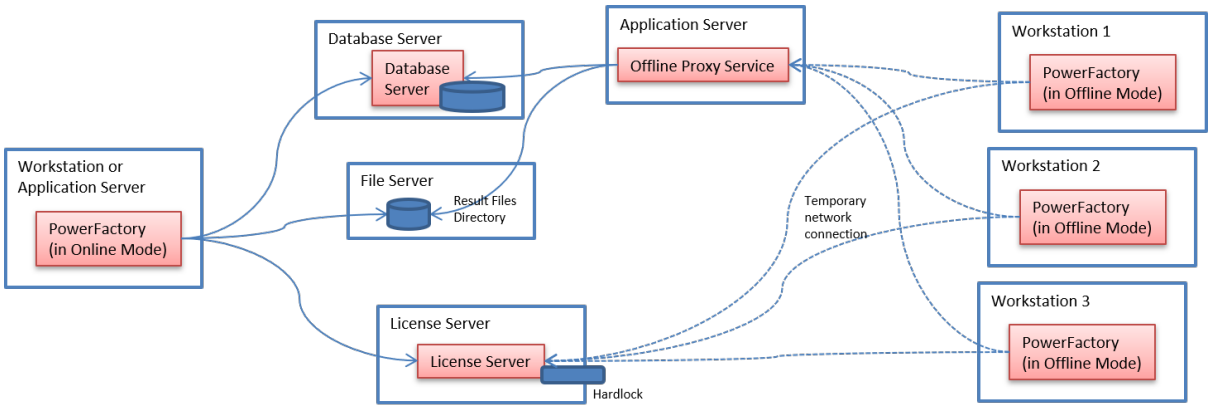

Figure 8.4.1: Offline Mode Components Overview

The installation procedure consists of the following steps:

1. First a *PowerFactory* environment in normal (i.e. not-offline) mode has to be set up. It contains at least a *PowerFactory* installation, the ***Licence Server***, and a multi-user database server (Oracle or SQL Server) (see section 8.4.1).
2. Installation of the ***Offline Proxy Service*** on an application server (see section 8.4.2).
3. Configuration of an Offline *PowerFactory* (see section 8.4.3).

A final section describes the procedure when upgrading to a newer *PowerFactory* version (see section 8.4.4).

8.4.1 *PowerFactory* in Normal Mode

Before *Offline Mode* can be set up *PowerFactory* must be installed and a multi-user database must be configured and initialised (see section 6.2 on page 21).

The **`Database`** page of the *PowerFactory Configuration* might resemble figure 8.4.2

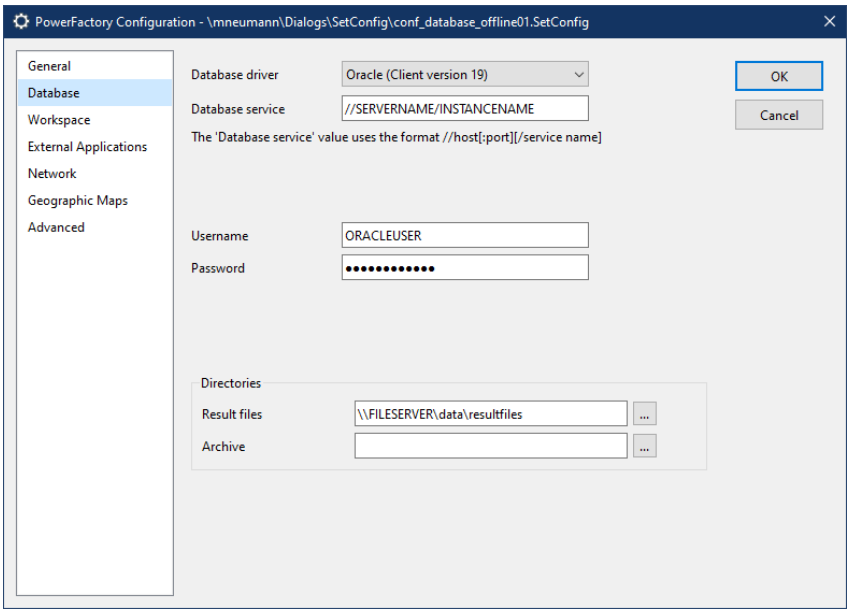

Figure 8.4.2: *PowerFactory* Database Configuration

Relevant for the ***Offline Proxy Service*** configuration are the parameters below:

- ***Oracle Database Server*** parameters
  - Connection specifier (e.g `//servername/instancename`)
  - Oracle user name (e.g. schemaname) and password
  - A network path to the *result files directory* (e.g. `\\fileserver\resultfiles`)
- ***SQL Server*** parameters
  - Connection specifier (e.g `servername\instancename`)
  - SQL Server user name (e.g. schemaname) and password
  - SQL Server database name
  - A network path to the *result files directory* (e.g. `\\fileserver\resultfiles`)

8.4.2 Offline Proxy Service

8.4.2.1 General Requirements

The ***Offline Proxy Service*** requires Microsoft .NET Framework 3.5.

8.4.2.2 Requirements for *Oracle*

Either ***Oracle Instant Client 11.2*** or a normal ***Oracle Client 11.2*** is required. The architecture must match the one of the ***Offline Proxy Service*** to be installed, either 32 Bit or 64 Bit.

Oracle Instant Client

To be able to access the Oracle Database you need to install a driver called Oracle Instant Client. It is also used with *PowerFactory* and can be obtained from the Oracle Homepage. The ***Offline Proxy Service*** requires version 11.2.

- Download the ZIP package from there.
- Unzip the package to the hard drive, preferably to `C:\oracle\instantclient_11_2`.

(Normal) Oracle Client

Install the Oracle Client using the installer.

8.4.2.3 Requirements for *SQL Server*

There are no additional requisitions.

8.4.2.4 *Offline Proxy Service* Installation

The ***Offline Proxy Service*** is installed using a Microsoft Installer (MSI) package. The installer files can be downloaded from our website. They are also shipped on the installation media, subfolder contents.

Both a 32 Bit and a 64 Bit version are available. The ***Offline Proxy Service*** architecture does not necessarily match the *PowerFactory* architecture, 32 Bit and 64 Bit can be mixed freely.

- Run the appropriate MSI installer and follow the on-screen instructions.

The ***Offline Proxy Service*** executable and configuration files are (for the 64 Bit version) by default located in the folder

```
C:\Program Files\DIgSILENT\PowerFactory Offline Service x.x\
```

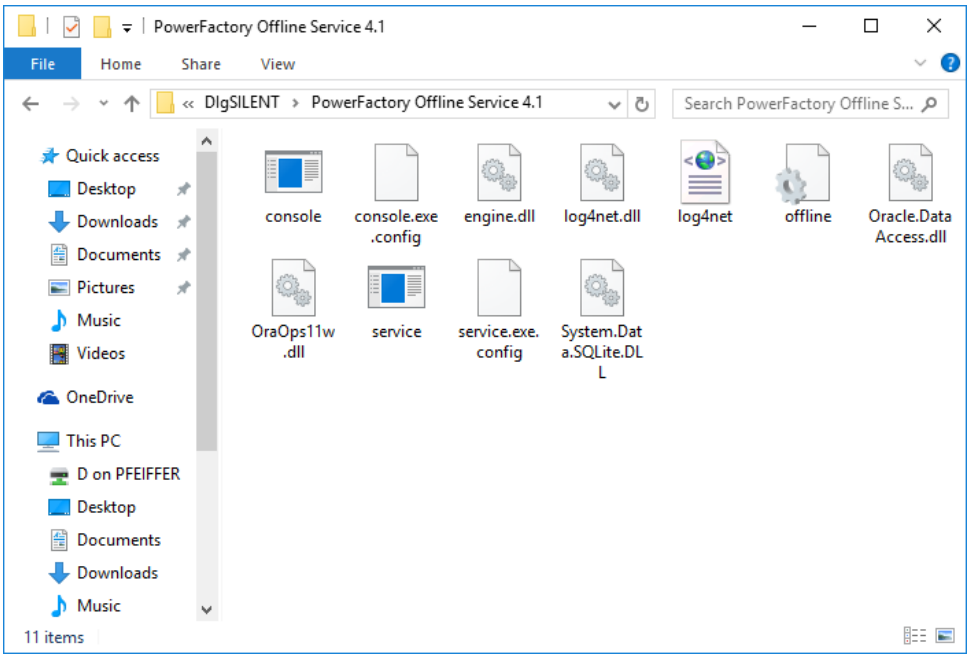

Figure 8.4.3: ***Offline Proxy Service*** Installation Directory

Before starting the ***Offline Proxy Service***, it must be configured.

8.4.2.5 Configuration file `offline.ini`

Navigate to the directory where you have installed the ***Offline Proxy Service*** and find the file `offline.ini`. Open it with any Text Editor, for example ***Notepad***. It will look like this:

```
[Network]
Port=9401

[Database]
Type=Oracle
DataSource=//servername/instancename
UserId=schemaname
Password=schemapassword
VaultPath=\\fileserver\resultfiles

[Folders]
DataRoot=E:\temp\offline
AdditionalPath=C:\oracle\instantclient_11_2
```

The different sections and their contents in detail:

Network Settings

```
[Network]
Port=9401
```

Port This port is opened by the server machine to accept incoming connections. It may be necessary to add an inbound rule to the Windows Firewall. This is covered later in this document.

Database settings for *Oracle*

```
[Database]
Type=Oracle
DataSource=//servername/instancename
UserId=schemaname
Password=schemapassword
VaultPath=\\fileserver\resultfiles
```

Type Must be set to `Oracle`

DataSource A string defining the server machine and Oracle instance to connect to.

UserId The Oracle user/schema id where the *PowerFactory* database schema is stored.

Password The encrypted (!) password for the Oracle user. Must be set using the `console.exe` using the `/setdbpassword` command from a command shell. Open a command window in the installation directory and type

```
console /setdbpassword <password>
```

VaultPath the *result files directory* by *PowerFactory*

These values should match the Database settings in the Online *PowerFactory* log on dialogue as described in section 8.4.1.

Database settings for *SQL Server*

```
[Database]
Type=SqlServer
Server=servername\instancename
UserId=sqlserverUsername
Password=sqlserverPassword
Database=sqlserverDatabase
VaultPath=\\fileserver\resultfiles
```

Type Must be set to `SqlServer`

Server A string defining the SQL Server machine and the instance name.

UserId The ***SQL Server*** user id

Password The encrypted (!) password. Must be set using the `console.exe` using the `/setdbpassword` command from a command shell. Open a command window in the installation directory and type:

```
console /setdbpassword <password>
```

Database the database name

VaultPath the *result files directory* network folder by *PowerFactory*

These values should match the Database settings in the Online *PowerFactory* log on dialogue as described in section 8.4.1.

Folder settings

```
[Folders]
DataRoot=E:\temp\offline
AdditionalPath=C:\oracle\instantclient_11_2
```

DataRoot This is a directory for temporary files created by the ***Offline Proxy Service***. It may be necessary to change the access rights on this directory. This is covered later in this document.

AdditionalPath The directories added here are temporarily added to the system PATH variable for this application only. This can be used to tell the ***Offline Proxy Service*** where to find the ***Oracle Instant Client*** libraries without modifying the systems PATH variable using Windows system settings.

Per default, the ***Offline Proxy Service*** will write log messages to the Windows application log. This behaviour is configurable through the file `service.exe.config`.

8.4.2.6 Setting up Security and Access Rights

The ***Offline Proxy Service*** is run as using the system `NETWORK SERVICE` user account.

- It is mandatory for the `NETWORK SERVICE` account to have read and write access to the directory specified as Folders/DataRoot folder. It is also mandatory to allow incoming connections on the port specified under Network/Port.
- It is mandatory for the `NETWORK SERVICE` account to have `read&write` access to the directory specified as Database/VaultPath
- It is mandatory for the `NETWORK SERVICE` account to have `read&write` access to the folder configured for logging `C:\Program Files\DIgSILENT\Offline Service`

- ***Oracle*** only: The `NETWORK SERVICE` account requires a `read` access to the ***Oracle Instant Client*** directory.

8.4.2.7 Firewall Settings

To allow connections from remote clients to the ***Offline Proxy Service***, the configured port must be added as Inbound Rule to the Windows Firewall. Open the Windows Firewall with Advanced Security MMC Snap-In. It can be searched for via the Start Menu in Windows 7 and above.

- Right click on **`Inbound Rules`** and select **`New Rule...`**
- Set **`Rule Type`** to `Port` and click **`Next`**
- Set **`Protocol`** to `TCP`. Enter the Port number that is configured in the configuration file for Network/Port and click **`Next`**.
- Select **`Allow the Connection`** and click **`Next`**.
- Check whatever is suitable for when this new rule is to be applied. This depends on your corporate network policy and click **`Next`**.
- Enter a **`Name`** and **`Description`** and click **`Finish`**.

Service Configuration in the MMC: No special actions needed. All general settings are working as usual.

8.4.2.8 Verify Configuration

Log on to the server as the Windows user, which will run the ***Offline Proxy Service*** as Windows service.

- Open a DOS console
- Change directory to the installation directory
- Run `console`

Now the ***Offline Proxy Service*** is started as a normal console application and writes messages into the console window:

```
C:\Program Files\DIgSILENT\Offline Service>console

2011-11-09 14:48:50,343 [1] INFO  Offline.Program - Acting as OfflineProxy Server.
2011-11-09 14:48:50,390 [1] DEBUG Offline.Engine - Engine.Start() ...
2011-11-09 14:48:50,781 [1] DEBUG Offline.Sockets.SocketServer -
    FileServerHandler('d:\tmp\offline_tmp') ...
2011-11-09 14:48:50,781 [1] DEBUG Offline.Sockets.SocketServer - ...
    FileServerHandler('d:\tmp\offline_tmp')
2011-11-09 14:48:50,781 [1] DEBUG Offline.Sockets.SocketServer -
    FileServerHandler('\\oracleserver\fullshared\offline_resultfiles') ...
2011-11-09 14:48:50,781 [1] DEBUG Offline.Sockets.SocketServer - ...
    FileServerHandler('\\oracleserver\fullshared\offline_resultfiles')
2011-11-09 14:48:50,781 [1] DEBUG Offline.Engine - ... Engine.Start()
Press Return to exit
```

Verify that no errors are reported.

- press **`Return`** key and close window

If there are any errors, adapt the configuration accordingly and restart `console` until there are no more errors.

8.4.2.9 Start the Offline Service

Having resolved all configuration errors ***Offline Proxy Service*** is ready to run as a Windows Service. Open Windows Services console window and start the Service.

- Right-click on `Offline Service`, and select `Start` in the context menu (see figure 8.4.4).

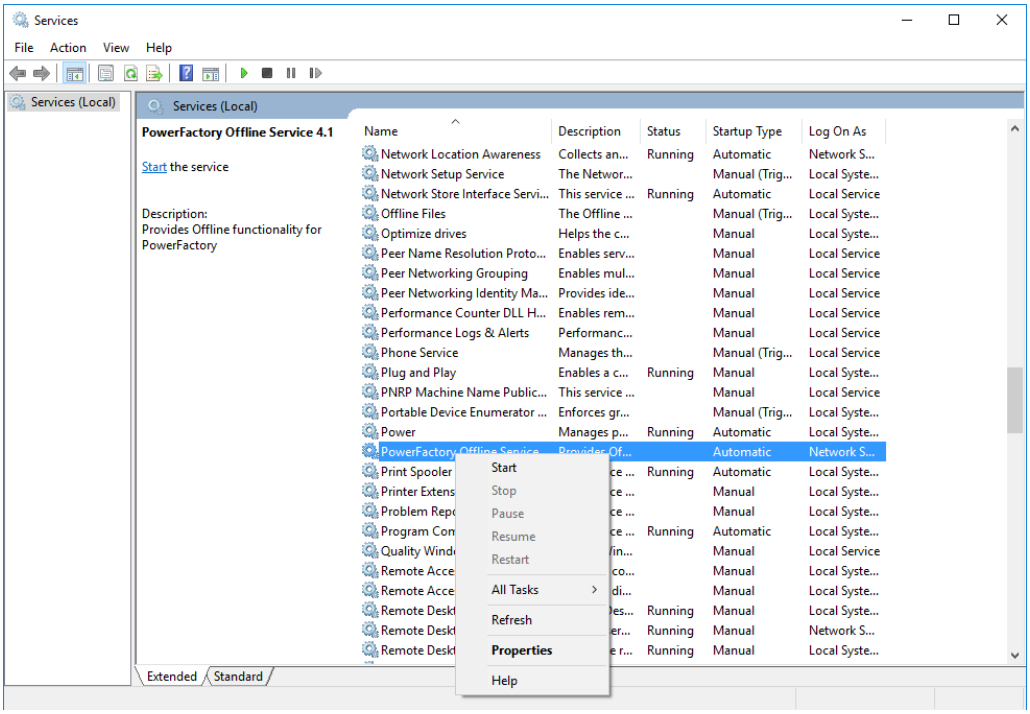

Figure 8.4.4: Windows Services

Adapt the service settings:

- Right-click on *Offline Service*, and select **`Properties`** in the context menu

An *Offline Service Properties* dialogue is shown.

- Set the Startup type to **`Automatic`**.

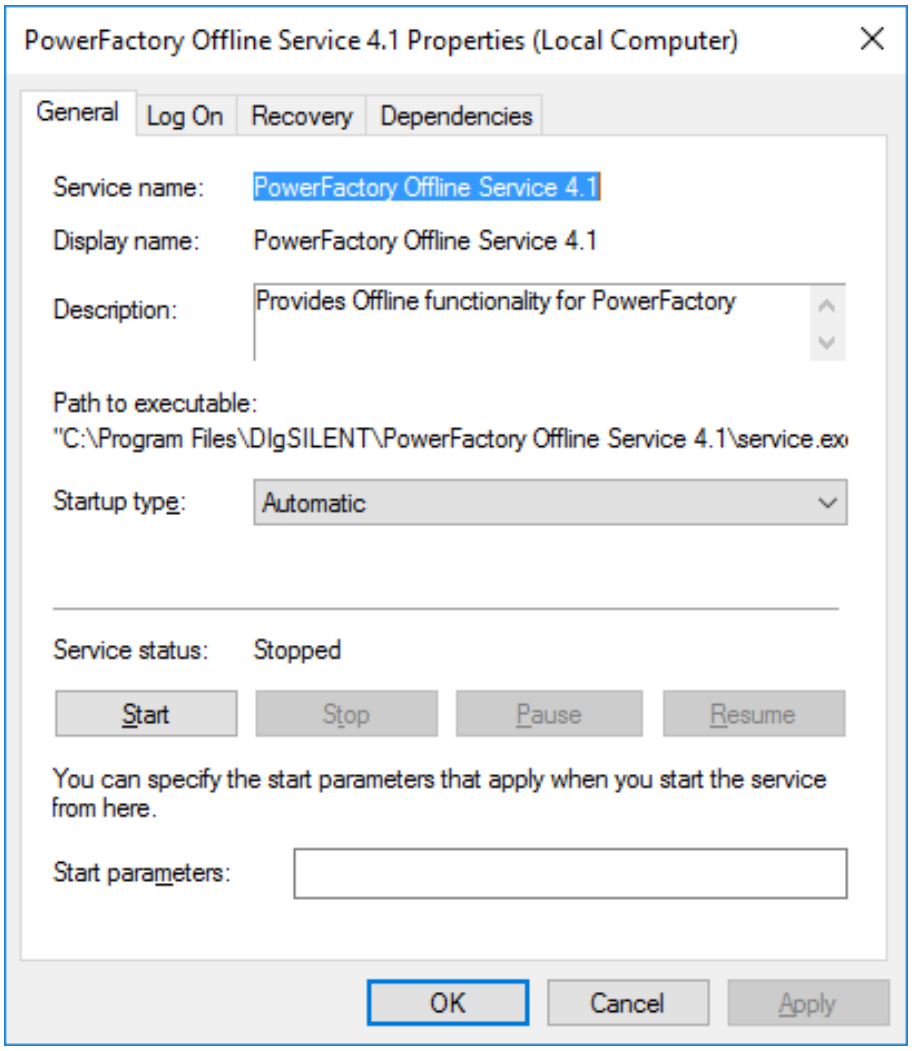

Figure 8.4.5: Offline Service Properties

- Eventually adapt the Windows user in the **Log On** tab.

The service writes messages into the log file
`C:\Program Files\DIgSILENT\Offline Service\service.log`

Before you proceed with the next section, verify that this log file was created and that there are no error messages in the log file.

8.4.3 *PowerFactory* Offline Mode Configuration

On each of the Workstations *PowerFactory* has to be configured in order to communicate with the ***Offline Proxy Service***.

Start *PowerFactory* in configuration mode (see section 8.6 on page 62).

- Switch to the **Database** page
- Insert the database connection settings as described below (see figure 8.4.6)

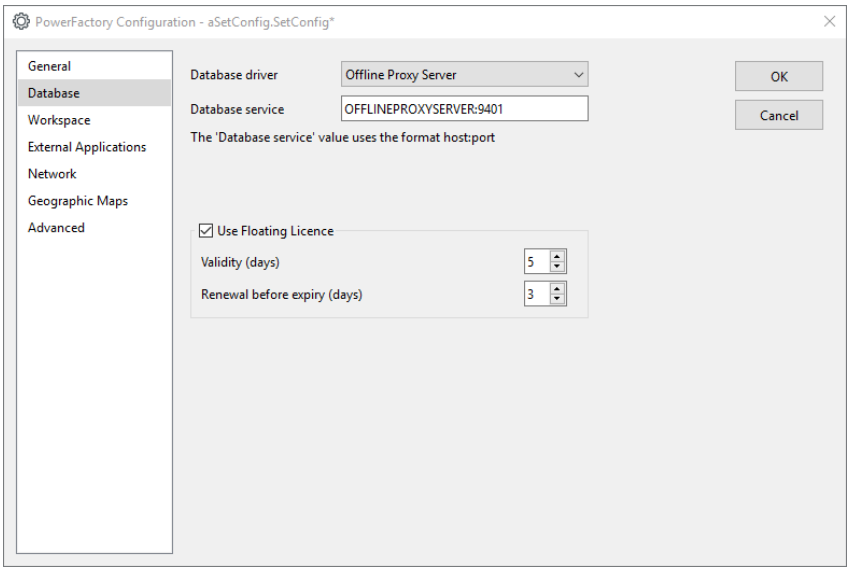

Figure 8.4.6: *PowerFactory* Configuration for Offline Mode

Database driver select **`Offline Proxy Server`**

Database service insert the host name or IP address of the Offline Server machine, followed by ":" and the port number e.g. `servername:9401` or `192.168.32.367:45600`

Use Floating Licence check this option if the offline mode should work without a permanent connection to the licence server (default).

- Validity: floating licences can be generated for a maximum of 30 days.
- Renewal before expiry: when getting close to expiration a floating licence can be renewed to extend its validity. In offline mode, *PowerFactory* reminds the user a configurable number of days before expiry and offers to start the renewal procedure (please note that renewal requires connection to the floating licence server).

8.4.4 *PowerFactory* Upgrade

This section describes the steps to take when *PowerFactory* is upgraded to a new version e.g. from `15.0.3` to `15.2.0`.

8.4.4.1 Step 0: Create Data Backups

Before changing the system create backups of

- ***Oracle*** Schema or ***SQL Server*** database used for *PowerFactory*
- the Vault directory on the Vault file server
- workspaces of all Offline *PowerFactory* instances

8.4.4.2 Step 1: Upgrade *PowerFactory* Online instance(s)

► Run the *PowerFactory* installer and replace the existing version with the new version.

8.4.4.3 Step 2: Migrate Online Database

With the upgraded *PowerFactory* installation it's possible to upgrade the database on the Database server.

- Start an upgraded online *PowerFactory* installation
- Confirm when asked for database migration.

8.4.4.4 Step 3: Upgrade Offline *PowerFactory* instances, migrate Offline Database

- Run the *PowerFactory* installer and replace the existing version with the new version.
- Start *PowerFactory*. The local offline database is migrated automatically.

8.4.4.5 Step 4: Upgrade *Offline Proxy Service*

- Make a Backup of all configuration files: `console.exe.config`, `offline.exe.config`, and `offline.ini`
- Stop the service
- Uninstall the service using the Add or remove programs tool
- Install the new service
- Restore the configuration files
- Restart the service

8.5 Administrator

Some functions (e.g. user management or changing the configuration in the *PowerFactory* database) require to start *PowerFactory* as *Administrator* user (*Administrator Mode*).

A separate short cut starts *PowerFactory* in *Administrator Mode*.

- Open Windows' Start menu and run *Windows Start button* → *All apps*→ *PowerFactory 2019*→ *PowerFactory (Administrator)*.

Alternatively it's possible re-start *PowerFactory* and log as *Administrator* via the menu *Tools* → *Switch User...*.

Usually the *PowerFactory Administrator* user has a password. Therefore you're asked to insert a password.

- Insert the password.
- Press **OK**.

8.6 Configuration

A separate short cut starts *PowerFactory* in *Configuration Mode*.

- Open Windows' Start menu and run *Windows button* → *All Apps*→ *PowerFactory 2019*→ *PowerFactory (Configuration)*.

A *PowerFactory Configuration* dialogue is shown. Alternatively you can review and change the configuration from within a running *PowerFactory* via the menu *Tools* → *Configuration...*.

The *Configuration* dialogue contains several pages which are explained in the next sub sections.

8.6.1 General Settings

See figure 8.6.1.

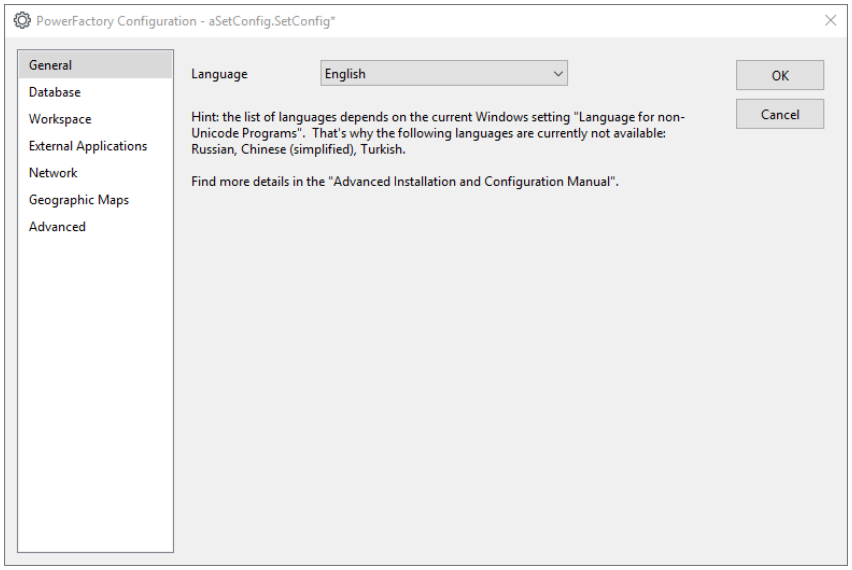

Figure 8.6.1: General Settings

Language Specifies the application language. Some language might require to adapt Windows' "Language for non-Unicode Programs" (see section 8.1 on page 46).

8.6.2 Database Settings

Depending on the database type there are different settings.

8.6.2.1 Local Database

See figure 8.6.2.

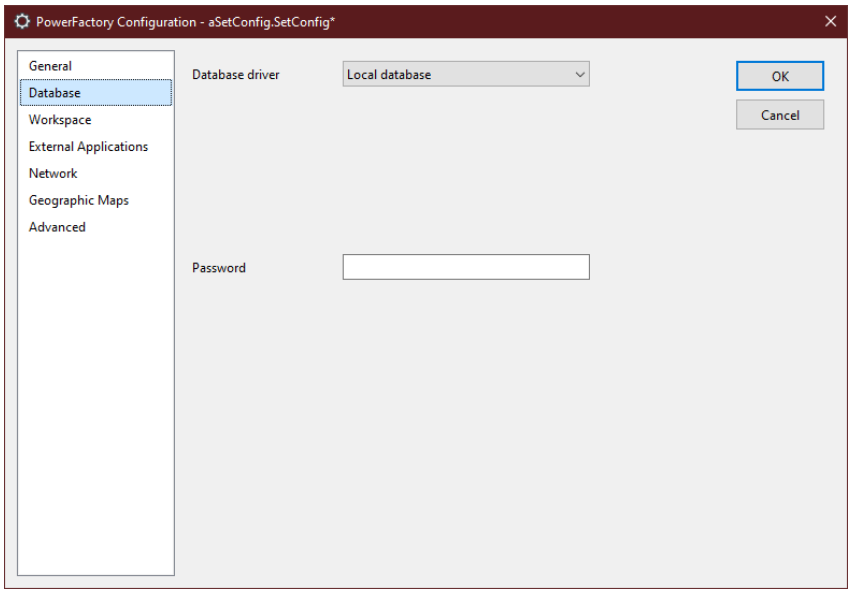

Figure 8.6.2: Database Settings (Local Database)

Password An optional password for the database encryption as described in section 6.1.2 on page 20.

8.6.2.2 Oracle native client

The **Database** page allows to set the Oracle connection settings (see figure 8.6.3).

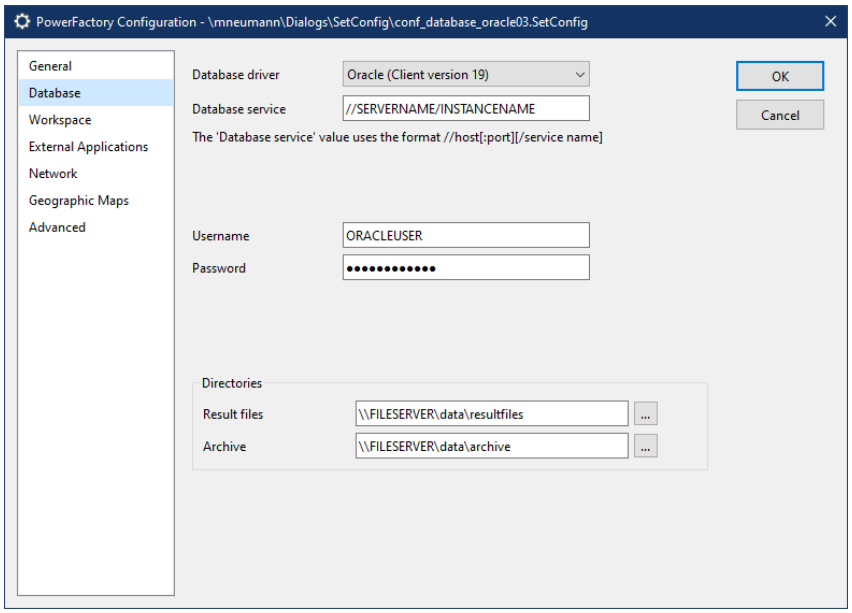

Figure 8.6.3: Database Settings (Oracle native client)

Database driver select `Oracle (Client Version 12c Release 1)` or `Oracle (Client Version 12c Release 2)`.

Database service this field describes the connection. It must be conform to the format

```
//host[:port][/servicename]
```

With the values used above (host=`oracleserver`, port=`1521` (default port), and SID=`PFSERVER`) the connection name is

```
//oracleserver/PFSERVER
```

If we had used a non-default port=`8888` the connection name would be

```
//oracleserver:8888/PFSERVER
```

If you've installed a (normal) Oracle Client and made an entry (e.g. `PFS`) in the `TNSNAMES.ORA` configuration file, you can use the TNS name instead. Then the **`Database service`** is just

```
PFS
```

Username and Password During the Oracle server setup an Oracle schema `PF` with the password `aPasswordForPf` has been created. Enter these values in the **`Username`** and **`Password`** fields.

Result Files/Archive Directory (Optional) Directories as described in section 6.2.4 on page 36.

Note: Usually the Oracle Client installation directory path must be configured on **`Advanced`** page. Otherwise *PowerFactory* can't find the required Oracle Client Runtime files.

8.6.2.3 Oracle ODBC client

The **`Database`** page allows to set the Oracle connection settings (see figure 8.6.4).

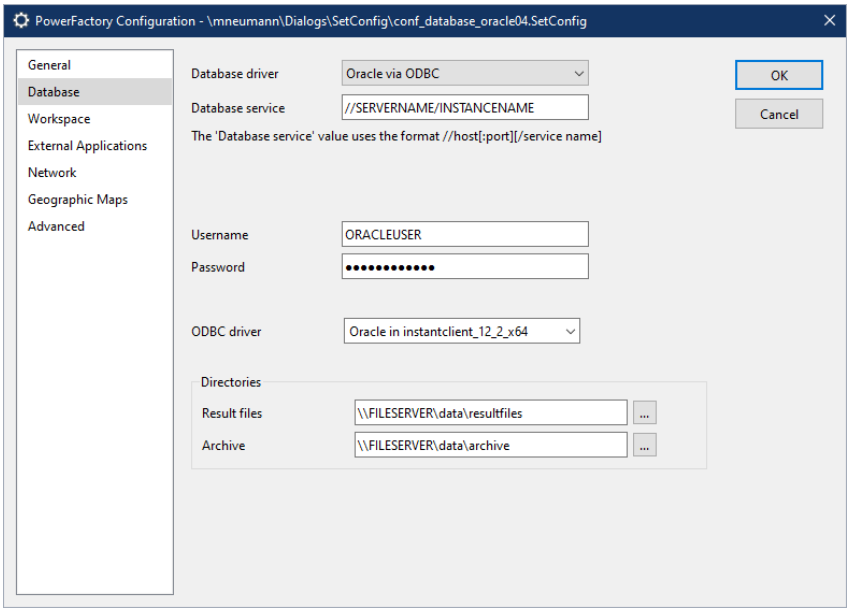

Figure 8.6.4: Database Settings (Oracle ODBC client)

Database driver select `Oracle via ODBC`

Database service, Username, Password, Directories see the Oracle native client section.

ODBC driver The name of the ODBC driver.

Note: Usually the Oracle Client installation directory path must be configured on **`Advanced`** page. Otherwise *PowerFactory* can't find the required Oracle Client Runtime files.

8.6.2.4 SQL Server

The **Database** page allows to set the SQL Server connection settings (see figure 8.6.5).

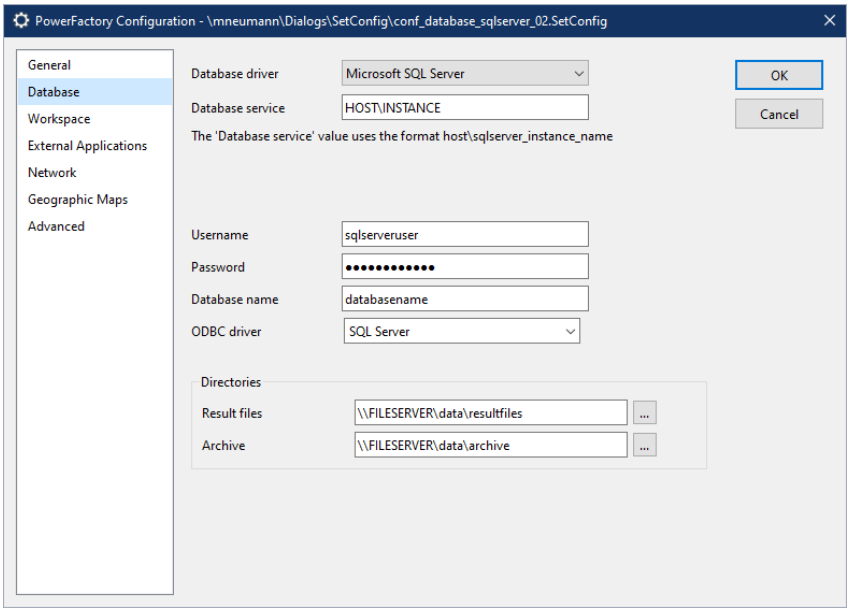

Figure 8.6.5: Database Settings (SQL Server)

Database driver Select `Microsoft SQL Server`

Database service The **Database service** uses the format

```
host\instancename
```

e.g.

```
MYSERVER\SQLEXPRESS
```

Username and Password The SQL Server name and password.

Database name The database name.

Result Files/Archive Directory (Optional) directories as described in section 6.2.4 on page 36.

8.6.2.5 Offline Proxy Server

PowerFactory's Offline Mode (section 8.4 on page 51) requires to configure a server.

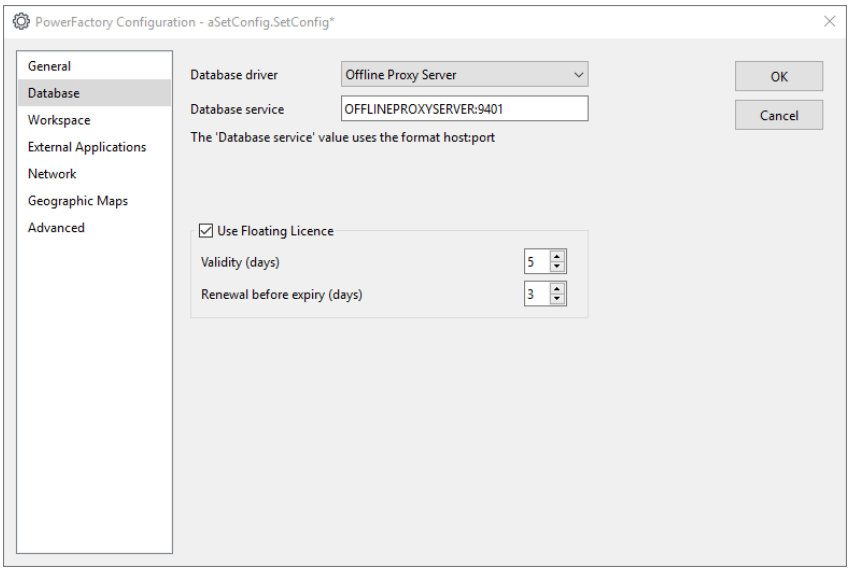

Figure 8.6.6: Database Settings (Offline Mode)

Database driver Select *Offline Proxy Server*

Database service Specify server host name and port, separated by a colon e.g.

```
OFFLINEPROXYSERVER:9401
```

Floating licence Configure floating licence usage. For details see section 8.4.3 on page 59.

8.6.3 Workspace Settings

The **`Workspace`** page allows to change the Workspace directories (see 8.6.7).

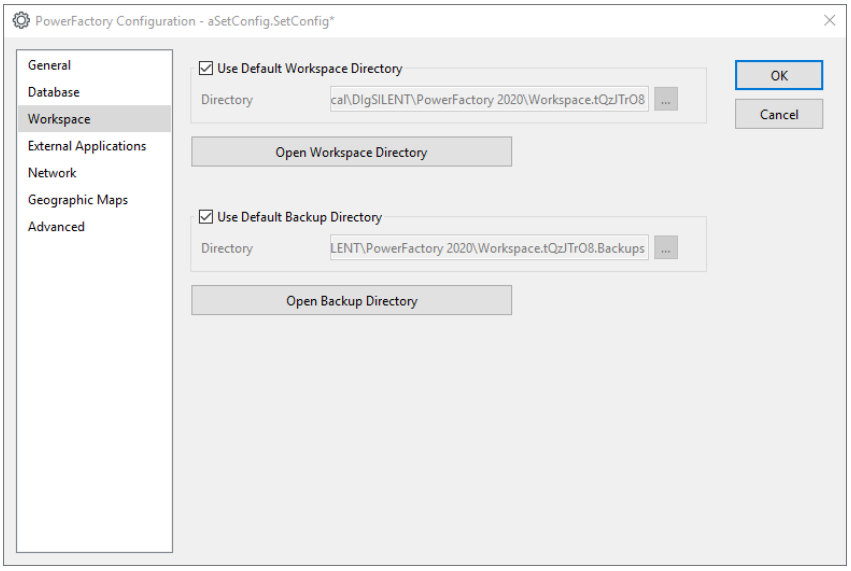

Figure 8.6.7: Workspace Settings

Use Default Workspace Directory Uncheck this option to specify a own Workspace directory.

Workspace Directory Path of the current Workspace directory.

Open Workspace Directory Opens Windows Explorer showing the Workspace directory.

Use Default Backup Directory Uncheck this option to specify a own Workspace Backup directory. This directory is used when workspaces are exported.

Backup Directory Path of the current Workspace Backup directory.

Open Backup Directory Opens Windows Explorer showing the Workspace Backup directory.

Note: Be careful when changing the **`Workspace Directory`**. The new workspace directory is not initialised with the current workspace directory. First export the workspace to a `*.zip` file and re-import it afterwards.

8.6.4 External Applications

The **`External Applications`** page allows to change the configuration when using external applications (see 8.6.8).

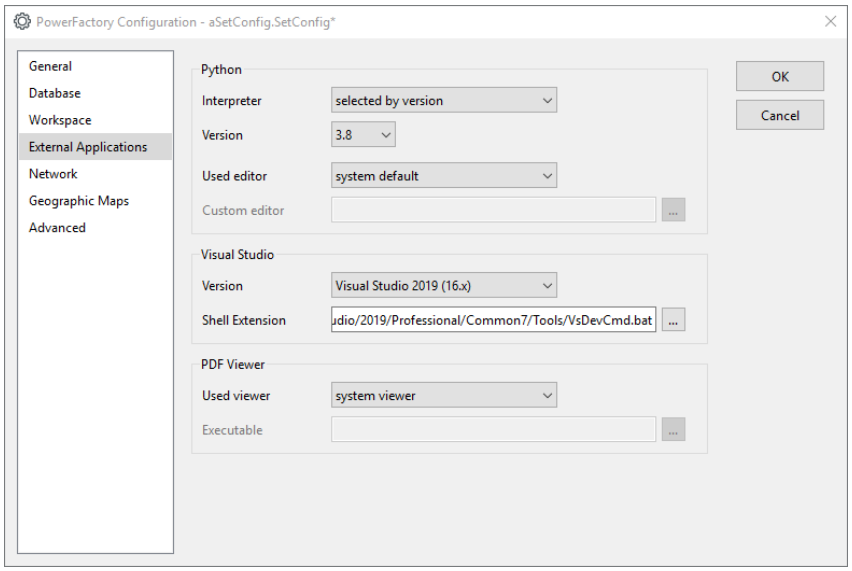

Figure 8.6.8: External Applications Settings

8.6.4.1 Python

Settings for running and editing Python scripts.

Version choose which Python version should be used. The Python version is expected to be installed.

Editor choose an application to edit Python script files

8.6.4.2 Visual Studio

Settings for compiling DSL models.

Version choose which Visual Studio version should be used. The Visual Studio version is expected to be installed.

Shell Extension Allows to choose a different Shell Extension

8.6.4.3 PDF viewer

PowerFactory's documentation is delivered as PDF documents. Here you can specify which application is used for displaying these documents

system viewer use the Window's default PDF viewer

SumatraPDF use the built-in viewer

custom insert the path to an arbitrary viewer application

8.6.5 Network Settings

Allows to configure general network parameters. These are used for loading background maps from Map servers and licence checks. It's possible to configure an HTTP proxy, optionally with authentication. See figure 8.6.9.

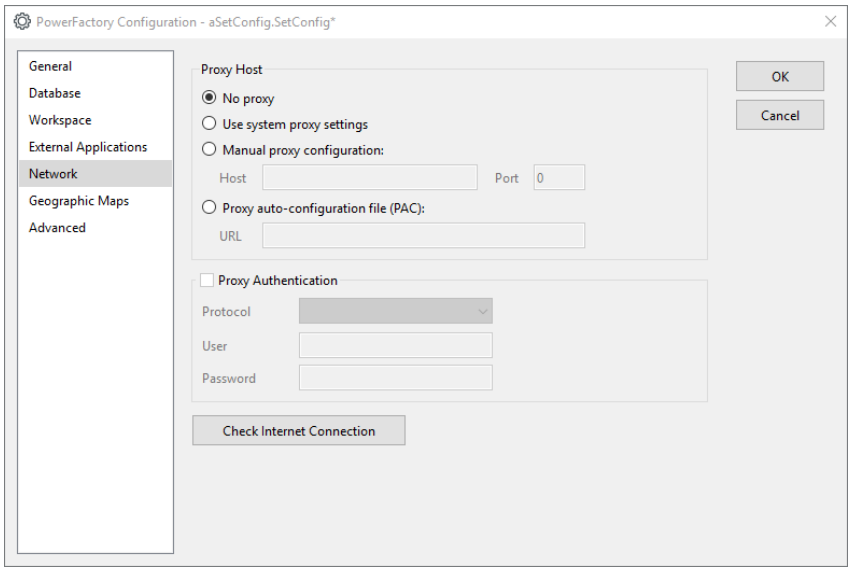

Figure 8.6.9: Network Settings

8.6.6 Geographic Maps

Settings for accessing Map Servers. See figure 8.6.10.

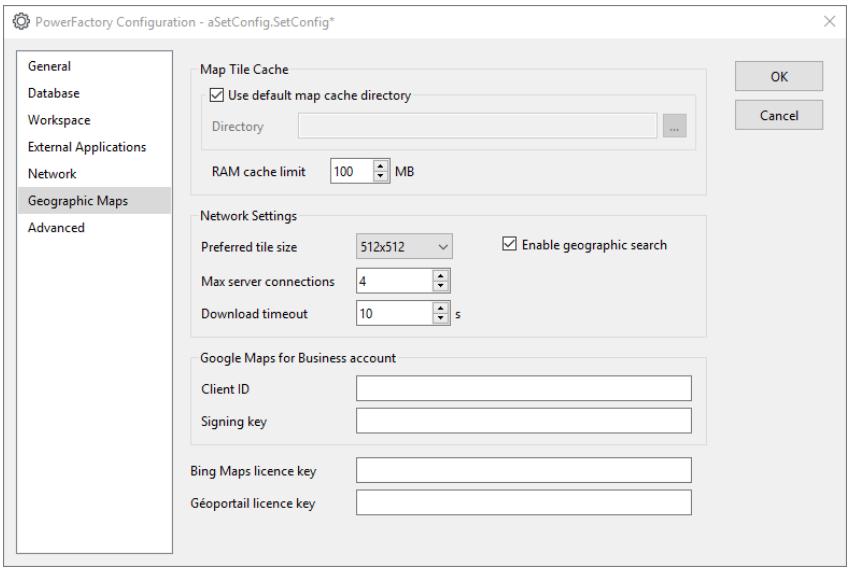

Figure 8.6.10: Geographic Maps Settings

Use default map cache directory Map data is downloaded from the Map Server and cached locally. You can specify a directory to share map tiles between users e.g. on a file server.

Preferred tile size The size of the fetched tiles

Max server connections Maximum number of concurrent downloads.

Download timeout Timeout used for server or network problems.

Google Maps/Bing Maps/Geoportail Access data for specific Map Server providers.

8.6.7 Advanced Settings

See figure 8.6.11.

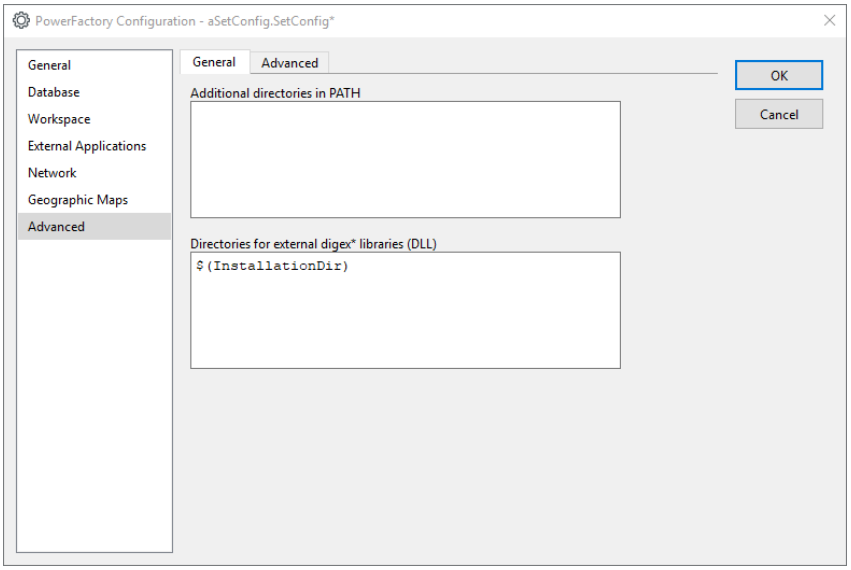

Figure 8.6.11: Advanced Settings

Additional directories in PATH A set of directories (each directory on a separate line) where *PowerFactory* should look for `*.dll` files e.g. the Oracle Client Runtime.

Directories for external digex libraries Set of paths that are searched for such DLL files.

The **Advanced** sub page allows to specify some really advanced settings (see figure 8.6.12).

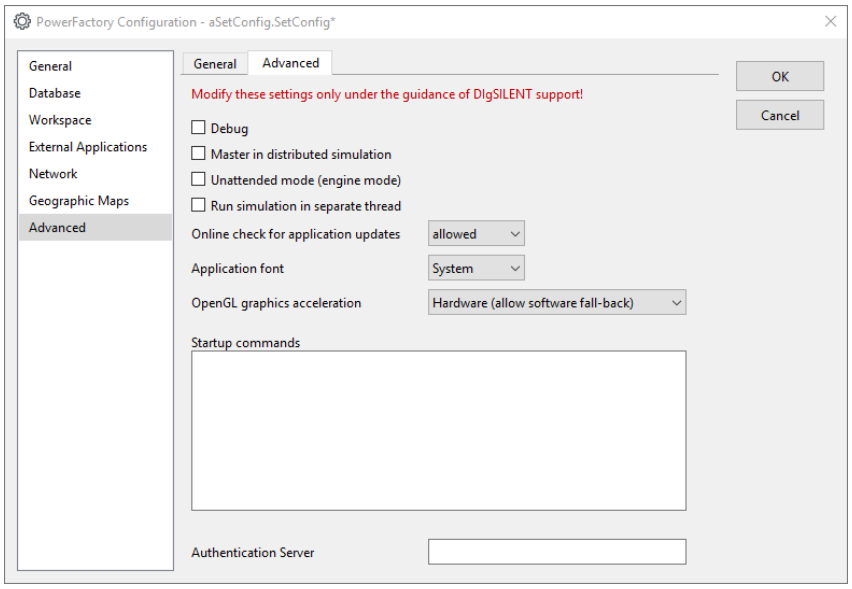

Figure 8.6.12: Advanced page, Advanced tab Settings

Debug Runs *PowerFactory* in Debug mode.

Master in Distributed Simulation If checked: this *PowerFactory* instance acts as Master, otherwise as slave.

Unattended Mode If checked: *PowerFactory* runs in non-interactive mode i.e. there no dialogues are shown which wait for user input.

Run simulation in separate thread Allows to run a simulation calculation a separate thread (experimental)

Startup Commands List of commands that are executed when *PowerFactory* is started.

Authentication Server Specifies an Authentication Server.

8.6.8 The `PowerFactory.ini` configuration file

The settings mentioned in the previous sections are persistently stored in the `PowerFactory.ini` file in the *PowerFactory* installation directory. Below find a typical configuration for a multi-user environment:

```
[advanced]
additionalPath = C:\Oracle\instantclient_12_2_x64

[database]
driver = oracle
password = 9pTOg5c/IjZochfLdBZ6373dJ
service = //oracleserver/instance
username = oracleschemaname

[general]
language = de

[license]
```

```
container = 130-1434393412
server = my.licenceserver.com
```

The command line parameter `/helpIni` displays a list of all settings (see section 8.7.2 on page 71).

It's possible to use Windows environment variables in the configuration e.g. `%USERNAME%` is replaced by the actual Windows user name.

8.7 Command Line Parameters

PowerFactory can be started with optional command line parameters.

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
"/parameter1:value" "/parameter2:another value"
```

Alternatively you can use the format below:

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
/parameter1 "value" /parameter2 "another value"
```

The double quote characters (`""`) can be omitted if the argument contains no spaces.

The remainder of this section describes only the most important parameters. Use `/help` to get a complete list of all parameters.

8.7.1 /help

Prints a list of all available command line parameters e.g.:

```
PowerFactory command line parameters:

/ini <path to INI file> - Path to a PowerFactory INI file

/username <name> - PowerFactory username

/password <password> - PowerFactory user password

/config - Edit PowerFactory configuration

[...]
```

8.7.2 /helpIni

Prints a list of all available parameters in the `PowerFactory.ini` configuration file.

```
[general] language=<language code> - application language, one of the
following values: "cn" (Simplified Chinese), "de" (German), "en"
(English), "es" (Spanish), "fr" (French), "ru" (Russian), or "tr"
(Turkish)

[general] title=<title> - text shown in application window

[general] workspacedirectory=<writable directory> - path to workspace
directory

[general] backupdirectory=<writable directory> - path to workspace
backup directory

[general] tempdirectory=<writable directory> - path to directory for
temporary files

[...]
```

8.7.3 /config

Show and edit *PowerFactory* configuration (see section 8.6 on page 62).

8.7.4 /housekeeping

Execute a database *Housekeeping* in the database (section 8.2.0.1 on page 47).

8.7.5 /ini

PowerFactory reads the configuration from the file `PowerFactory.ini` in the installation directory. In some scenarios it might be convenient to have several configuration files `PowerFactory_1.ini`, `PowerFactory_2.ini` etc. These files can be in any directory, not necessarily in the installation directory. For each configuration follow the procedure below.

1. Edit and save *PowerFactory* configuration (see section 8.6 on page 62). The configuration is saved to the `PowerFactory.ini` in the installation directory
2. Copy `PowerFactory.ini` to e.g. `C:\PowerFactory Configurations\PowerFactory_1.ini`
3. Start *PowerFactory* with the `/ini` parameter:

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
    /ini "C:\PowerFactory Configurations\PowerFactory_1.ini"
```

The path to the `*.ini` file must be an absolute path. A leading dot (`.`) is interpreted as *PowerFactory* installation directory. Example:

```
/ini ".\PowerFactory_1.ini"
```

is equivalent to

```
/ini "C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory_1.ini"
```

8.7.6 /killFormerSession

If there is already an active session of the *PowerFactory* user, it will be terminated without any confirmation. This parameter is applicable for multi-user databases only.

8.7.7 /lang

`/lang <LANGUAGE>` Ignore the **Language** setting in the configuration, and run *PowerFactory* with a different language:

- `cn`: Simplified Chinese,
- `en`: English,
- `es`: Spanish,
- `de`: German,
- `fr`: French,
- `ru`: Russian, or
- `tr`: Turkish

8.7.8 /minimalMigration

Perform a minimal migration of the database (see section 9.3.3 on page 83 and section 9.3.4 on page 85 for details).

8.7.9 /migration

Migrate all not-migrated projects after a Minimal Database Migration (see section 9.3.3 on page 83 for details)

8.7.10 /readonlymode

Database read-only mode (see section 6.3 on page 37 for details).

8.7.11 /inMemory

Database in-memory mode (see section 6.1.3 on page 20 for details).

8.7.12 /username and /password

`/username:<USERNAME>` specifies which *PowerFactory* user is to be used. Example: start as *Frodo Baggins* user.

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
/username "Frodo Baggins"
```

Optionally a password can be specified with `/password:<PASSWORD>`:

```
"C:\Program Files\DIgSILENT\PowerFactory 2019\PowerFactory.exe"
/username "Frodo Baggins" /password "Frodos password"
```

8.7.13 /unlockdatabase and /resetunlockkey

Use previously set database unlock key (DUK) or reset the DUK to a new value. When starting *PowerFactory* with one of these options a dialogue is shown asking for the currently configured DUK / the *PowerFactory* administrator password and guide the user through the selected process (see section 7.4.1 on page 44 for details).

8.7.14 /reportLogFile

Report log file paths in console.

8.7.15 /createSupportPackage

```
/createSupportPackage <filename>
```

Creates a support package with the specified file name.

8.7.16 /resetWindowPosition

Resets the position, size and layout of the *PowerFactory* main window.

8.7.17 /general.workspacedirectory

```
/general.workspacedirectory <directory>
```

Specifies the path to the workspace directory.

8.7.18 /database.driver

```
/database.driver <driver>
```

- `sqlite`: Local database
- `oracle`: Oracle database using Oracle 12 Release 1 via OCCI
- `oracle122`: Oracle database using Oracle 12 Release 2 via OCCI
- `oracleodbc`: Oracle via ODBC
- `sqlserver`: Microsoft SQL Server
- `offline`: Offline mode

8.7.19 /database.service

```
/database.service <service>
```

Database service endpoint, depending on the database driver used.

8.7.20 /database.username

```
/database.username <username>
```

Username to be used when connecting to the database service.

8.7.21 /database.password

```
/database.password <password>
```

Encrypted password for database service.

8.7.22 /database.database

```
/database.database <identifier>
```

Database or schema identifier, depending on database driver used.

8.7.23 /database.odbcDriver

```
/database.odbcDriver <driver>
```

ODBC driver to be used for the ad-hoc connection. Only applies to SQL Server or Oracle via ODBC.

8.7.24 /database.resultFilesDirectory

```
/database.resultFilesDirectory <directory>
```

Path to the *result files directory*.

8.7.25 /database.archiveDirectory

```
/database.archiveDirectory <directory>
```

Path to the *archive directory*.

8.7.26 /license.server

```
/license.server <server>
```

Name or network address of the network licence server.

8.7.27 /license.container

```
/license.container <container>
```

CodeMeter licence container Id.

8.7.28 /license.hotstandbyserver

`/license.hotstandbyserver <server>`

Name or network address of the hotstandby network licence server.

8.7.29 /license.hotstandbycontainer

`/license.hotstandbycontainer <container>`

CodeMeter licence container Id of the hotstandby licence server.

8.7.30 /advanced.additionalPath

`/license.hotstandbycontainer <directory1;directory2;directory3>`

Additional directories in PATH, separated by semicolon.

8.7.31 /advanced.commands

`/advanced.commands <command1;command2;command3>`

Additional commands to be executed after logging in to PowerFactory, typically used to execute DPL commands after startup.

8.8 Silent Installation Options

The *PowerFactory* installer can be run in *silent mode* via the command line. The easiest way to trigger a silent installation of *PowerFactory* is by executing the command

```
PowerFactory-2019_x64.exe -quiet -install
```

in the directory where the *PowerFactory* installation executable is located.

Please note that this command will execute the installation in a new process, no graphical user interface will be shown and no other interaction with the installation process is possible. Per default the program will install all features and packages of the product to the default installation location `%PROGRAMFILES%\DIgSILENT\PowerFactory 2019\`.

Note: Administrator privileges are required to successfully run the installation. The silent installation will be executed in a separate process and after issuing an installation command the command line will immediately return. The installation process continues to run in the background until it finishes or is cancelled due to an error (see subsection 8.8.2).

8.8.1 Command line options

To start any silent installation operation, type the name of the *PowerFactory* installation executable, the argument `-quiet` and one of the following *operation arguments*:

`-install`	installs *PowerFactory*.
`-uninstall`	removes an existing PowerFactory installation. If no PowerFactory installation is present, this command will have no effect.

Figure 8.8.1: Operation commands for silent installation

The installer supports the following additional command line arguments when using `-install`:

- `INSTALLDIR`: Specifies the installation directory. Must be a full path where the application should be installed. The installer will not create a sub folder for the product.
- `CREATEDESKTOPICON`: Enables or disables the creation of a shortcut for the application on the user's desktop. Possible values are "yes" or "no". The default value is "yes".

Note that the following rules must be satisfied for any chosen installation path:

- the installation path defined by the `INSTALLDIR` parameter must be given in quotes and may **not** end with a backslash (`\`). If you want to install *PowerFactory* into the directory `E:\PF 2019`, for instance, use `INSTALLDIR="E:\PF 2019"`. Do **not** use `INSTALLDIR="E:\PF 2019\"` or `INSTALLDIR=E:\PF 2019`.
- The installation path must be given as an absolute path to the desired installation folder (e.g. `D:\path\to\folder`, **not** `\relative\path` or `D:\path\to\file.xyz`).
- The corresponding drive of the chosen path
  - must exist,
  - must **not** be a read-only drive (such as a DVD drive),
  - must **not** be a network drive
  - and must have sufficient free disk space.
- The administrator of your system needs reading and writing rights for the installation path.
- The installation folder must be empty.

During installation or removal, log files are created. By default they are written to `%LOCALAPPDATA%\Temp\PowerFactory*.log`. However, you may specify a custom log file location by using the `-log` command argument:

```
PowerFactory-2019_x64.exe -quiet -install -log "D:\path\to\log-file.txt"
```

Besides the main log file, additional log files for the separate packages of the installation bundle are created in the same directory and use the name of the main log file as prefix.

8.8.2 Error codes

Whenever the installation operation finishes or is cancelled, an error code is returned to indicate what went wrong (if something went wrong).

The error code of the operation will be written into the log file *as hexadecimal number* and it may be easily found by searching the log text for the term '`Exit code: 0x`'. For instance, the output `Exit code: 0x1002` means that the operation yielded the error code **0x1002** (which is 4098 in the decimal system).

Please note: The error code **0x0** indicates that the operation was successful, i.e. there were no errors.

There are three types of error codes that may be returned by the installer:

- **Windows Installer Error codes**: These error codes are transmitted by the *Microsoft Windows Installer* engine. Any error code smaller than 4096 (hexadecimal: 0x1000) indicates a *Windows Installer error*. For an explanation for these error codes, please refer to the official Microsoft Developer Network (*MSDN*) pages, e.g. https://msdn.microsoft.com/en-us/library/windows/desktop/aa368542(v=vs.85).aspx or https://msdn.microsoft.com/en-us/library/windows/desktop/ms681385(v=vs.85).aspx.
- ***PowerFactory* installer error codes**: These error codes indicate errors specific to the *PowerFactory* installer. The codes start at 4096 (hexadecimal: 0x1000) and are listed in figure 8.8.2.
- **MSI error code**: In some cases, one of the *Microsoft Installer* (MSI) packages included in the *PowerFactory* installer will abort an operation. In that case, the returned error code will be 8192 (hexadecimal 0x2000). The specific nature of the error is given in the accompanying log file.

Error code	Explanation
0x1000	The operation was cancelled by the user.
0x1001	Not enough space on operating system drive. Your operating system's root drive (typically `C:\`) does not have enough free disk space for installation. Please note that regardless of the chosen installation path, the installation also requires disk space on the operating system drive.
0x1002	Not enough space on target installation drive. The root drive of the chosen installation path does not have enough free disk space.
0x1003	The selected installation folder is not empty.
0x1004	The chosen installation path is invalid. Note that the installation path must be given as an absolute path to the desired installation folder.
0x1005	The chosen installation path is on a network drive. Note that *PowerFactory* cannot be installed on a network drive.
0x1006	The root drive of the installation path does not exist or does not allow installation.
0x1007	Access to the installation path was denied. Note that installation requires administrator privileges and the administrator needs to have reading and writing rights for the selected installation path.
0x1008	A 64 bit version of the installer is being executed on a 32 bit operation system.
0x1009	Administrator rights were not granted for the installer. Note that installation requires administrator privileges.

Figure 8.8.2: Comprehensive list of *PowerFactory* installer error codes

8.8.3 Examples

To install *PowerFactory* silently to the directory `E:\PowerFactory 2019`, use the command

```
PowerFactory-2019_x64.exe -quiet -install INSTALLDIR="E:\PowerFactory 2019"
```

To remove an existing *PowerFactory* installation and create the log file `D:\temp\log.txt`, use

```
PowerFactory-2019_x64.exe -quiet -uninstall -log "D:\temp\log.txt"
```

Chapter 9

Upgrade and Migration

This chapter addresses typical scenarios where an existing *PowerFactory* installation is modified.

- Licence Migration: an existing licence is migrated (see section 9.1 on page 79)
- Upgrade to a newer *PowerFactory* version e.g. from ***15.2*** to ***2019*** (see section 9.2 on page 81).

9.1 Licence Migration

With the release of *PowerFactory* 2016 the licensing system has been re-implemented and is based on a new technology since. Users of former versions of *PowerFactory* (15.x or previous), need to migrate their old licence when upgrading to *PowerFactory* 2016 or later. Please visit: `https://www.digsilent.de/en/licence-migration.html`.
After you have received the Activation Key of your migrated *PowerFactory* licence from *DIgSILENT* sales, please activate it as described in Section 5.3.

9.1.1 Using *PowerFactory*15.2 or earlier with a *PowerFactory*2019 Licence

Users who have already migrated their *PowerFactory* licence but nevertheless need to run earlier versions of *PowerFactory* (15.x or previous) from time to time, will have to install a new version of the *PowerFactory* licence server, which is called `LegacyLicenceService`.

Hint: The `LegacyLicenceService` requires a *local* licence (dongle connected to the local machine or softkey activated on the local system). *PowerFactory* instances however can connect to the `LegacyLicenceService` from any PC in the network.
Also note that - due to changes in the functionality contained in *PowerFactory* base package from *PowerFactory* 15.x to *PowerFactory* 2016 and later - it is only possible to run an older *PowerFactory* version with a recent *PowerFactory* licence that contains at least the following additional modules (which are contained in any migrated licence): Contingency Analysis, Quasi-Dynamic Simulation, Network Reduction, Techno-Economical Analysis, Scripting and Automation.

9.1.1.1 Installation of Legacy Licence Service

The installer for `LegacyLicenceService` can be found in the *PowerFactory* 2019 installation directory (folder `Legacy Licence Service`). After running the installer, the *Licence Service Utility* tool can be started by:

- Opening the Windows' Start menu and running *Start* → *All Programs*→ *PowerFactory Legacy Licence Service*→ *Licence Service Utility* or
- Running the `LicenceServiceUtility.exe` in the *Legacy Licence Service* installation directory.

Hint: This tool can only be started if a migrated *PowerFactory* licence is available on the local machine.

Within this tool press the **`Install Legacy Licence Service`** button. If there are different *PowerFactory* licences available on the local computer, it is important to select one licence before starting the service (see 9.1.1.2).

After installing, `LegacyLicenceService` will start automatically when rebooting your system. However, `LegacyLicenceService` depends on the start of CodeMeter. To configure this dependency, open a command prompt with administrator rights and type:

```
sc config LegacyLicenceService depend= CodeMeter.exe
```

9.1.1.2 Configuration and start of *PowerFactory*

Configuration for *PowerFactory* Workstation Licence

If using a *PowerFactory* workstation licence, *PowerFactory* has to be configured as workstation. Run `TOOLS\Configure.bat` in the installation directory of your old *PowerFactory* version. The *PowerFactory* configuration dialogue opens. Navigate to page *Licence* and select *PowerFactory Workstation*. Close the dialog.

For using a workstation licence it is important to stop the old *PowerFactory* licence service (DIgLiseService) on the local system (if all your licences have been migrated to the new licence system you can uninstall the service.).

Configuration for *PowerFactory* Network Licence

If using a *PowerFactory* network licence, *PowerFactory* has to be configured as server. Run `TOOLS\Configure.bat` in the installation directory of your old *PowerFactory* version. The *PowerFactory* configuration dialogue opens. Navigate to page *Licence* and select *PowerFactory Server*. In the *Server Name* field enter `127.0.0.1` or the IP address of the PC where `LegacyLicenceService` is running. Go to the *Advanced* tab and change the RPC-Endpoint to *4010*. Close the dialog.

Selection of a specific *PowerFactory* Licence

If there are more than one migrated *PowerFactory* licences available on the local system, it is important to configure the licence to be used by pressing the button **`Select Licence Container`** in *Licence Service Utility*. This will open the **`Select Licence`** page of the ***Licence Manager*** which is described in 5.5.

Afterwards the file `LegacyLicenceService.ini` (located in the installation directory) should contain a `[licence]` section similar to the following:

```
[license]
container = 128-8130889
```

Hint: Licence configuration takes effect after restarting the LegacyLicenceService.

Start *PowerFactory*

Make sure *Legacy Licence Service* is running. It can be started from the *Licence Service Utility* tool (see 9.1.1).

Then start your old *PowerFactory* version as usual.

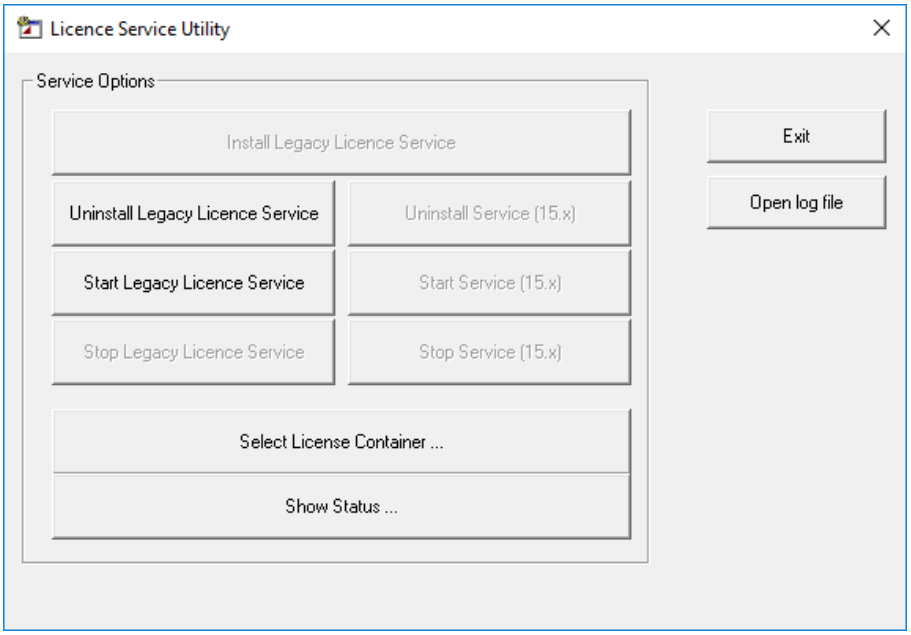

Figure 9.1.1: Licence Service Utility

9.2 Upgrade *PowerFactory* Version

Beginning with *PowerFactory* 2016, upgrade installations are no longer supported. Each PowerFactory release must be installed as a separate, new product.

9.3 Data Migration

A data migration is required when switching from one major release to another one, e.g. from version 2018 to version 2019. There is no migration required if only the service pack number changes. In the latter case, only the following folders are updated:

- System
- Library

Please make sure these folders do not contain any custom data. They are reset in the process.

9.3.1 Local Database

Since version *PowerFactory* 15.1 workspaces (including the local database) can be easily exported and imported.

- Start the former *PowerFactory* and export the workspace to a `*.zip` file (section 6.1.1.1 on page 19)
- Start the new *PowerFactory* and import the `*.zip` file.

9.3.2 Multi-User Database

Since version 14.0 *PowerFactory* can access and use multi-user database from former versions. The database is migrated automatically when the new *PowerFactory* version is started. The *PowerFactory Administrator* password is required.

Note: A multi-user database is always migrated in-place. After migration the former *PowerFactory* version won't be able to use the database anymore. Ensure that the former version is disabled or completely removed.

Note: The database migration temporarily requires more data space for e.g. intermediate tables or table indexes which are dropped at the end of the migration. On Oracle ensure that the temporary tablespace `TEMP` can grow up to at least 10 percent of the size of the `OBJECT_` table.

The migration resets all changes in these top-level objects:

- System
- Library

Before you migrate, please make sure that these folders don't contain any data you need afterwards.

- Inform all *PowerFactory* users about the migration.
- (Optional) Disable the former *PowerFactory* version (e.g. on Application Server).
- Ensure that no *PowerFactory* user is logged on.
- Create a database backup.
- The new *PowerFactory* installation must be configured to use the same database connection parameters.
- Start the new *PowerFactory* version. A warning dialogue is shown (see figure 9.3.1).

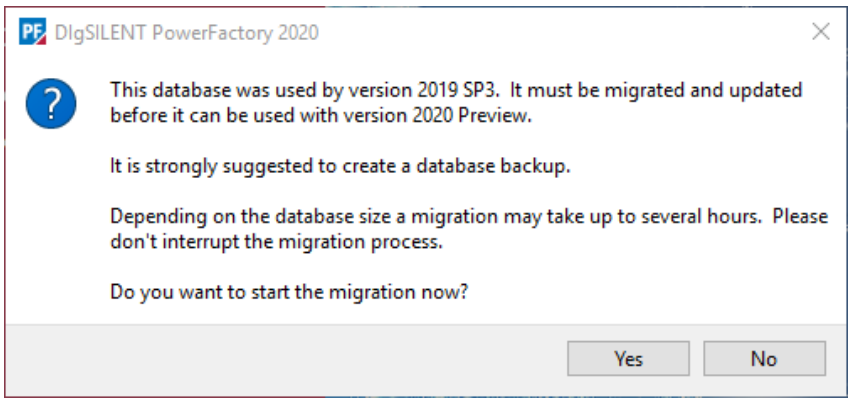

Figure 9.3.1: *DB Migration* Dialogue

- Press **`Yes`** to start the migration.

Before the migration is started you're asked for the *PowerFactory Administrator* password.

The existing database is now migrated to the new *PowerFactory* database structure. Depending on the size of the database this may take several minutes up to several hours. Please don't interrupt the migration process. When the migration is completed a success dialogue will appear (see figure 9.3.2).

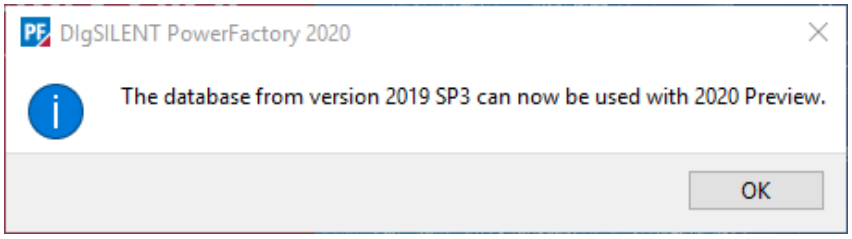

Figure 9.3.2: Successful Database Migration

9.3.3 Complete vs. Minimal Database Migration

Earlier *PowerFactory* versions always migrated the database completely. A complete migration—especially on a multi-user environment—could run for several days depending on the database size (i.e. the number of users, the number of projects, and the project sizes) and the available hardware resources. *PowerFactory* users could not use the application during that period.

Since *PowerFactory* version 15.2 it's possible to run a *Minimal Migration*, that reduces the downtime period for big database essentially. It only alters the database structure, but doesn't migrate the users' projects.

Before the actual migration is started a dialogue allows to choose between `Complete` and `Minimal` migration (see figure 9.3.3).

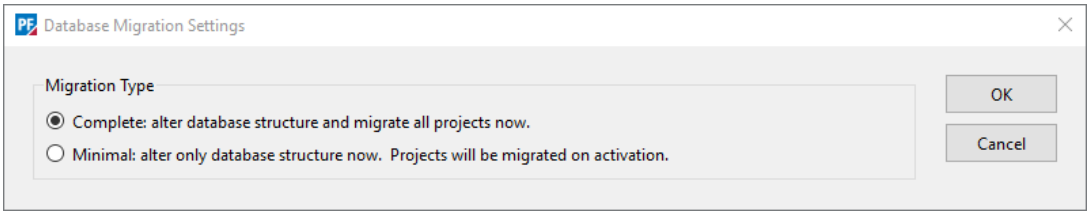

Figure 9.3.3: *Database Migration Settings* Dialogue

- **`Complete`** (recommended): alters database structure and migrates all projects right now. This may take very long depending on the number of projects and their sizes.
- **`Minimal`**: alters only the database structure. Projects will be migrated later on first activation.

After a Minimal Migration the Data Manager displays not-migrated projects in a grey-coloured font without any content (see figure 9.3.4). Not-migrated can be renamed, moved, and deleted like normal projects. They're automatically migrated on activation.

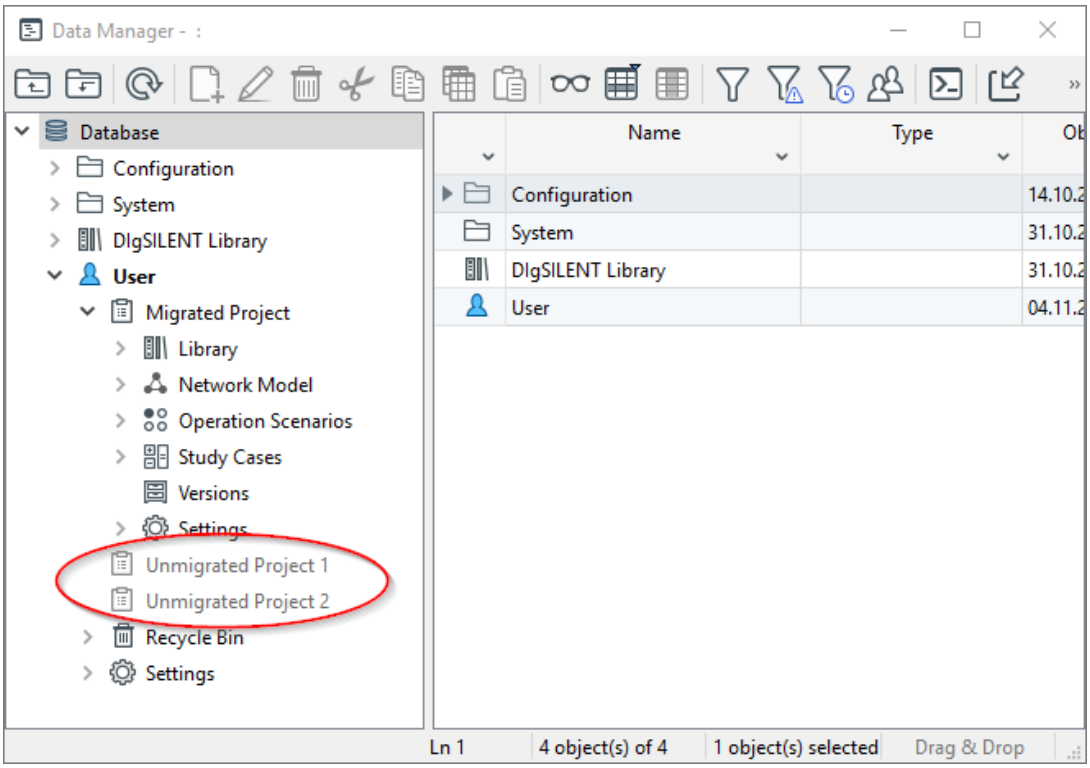

Figure 9.3.4: Not-migrated project

Some projects might not be used any more, and stay un-migrated forever. However, in some cases it might be desirable to enforce the migration of the projects. On account of this *PowerFactory* can be started in *Migration Mode* with the `/migration` command line argument:

```
PowerFactory.exe /migration:<maximum duration in hours>[:<sleep interval in seconds>]
```

It migrates not-migrated projects sequentially and stops either after a given time period has passed, or there aren't any more projects to migrate.

```
PowerFactory.exe /migration:8
```

runs for up to 8 hours. An optional parameter allows to set a sleep time between migrating two projects in order to reduce the load on the database server. The command below runs the migration for up to 8 hours, pausing for 60 seconds after each project.

```
PowerFactory.exe /migration:8:60
```

The migration order of the projects can be influenced by the *PowerFactory* users. A relative migration priority can be set in the project dialogue (see figure 9.3.5). First all projects with a `High` priority are migrated, then all projects with `Medium` priority, and finally projects with `Low` priority. Projects with the most recent activation date are favoured. Base projects are automatically migrated before their derived projects.

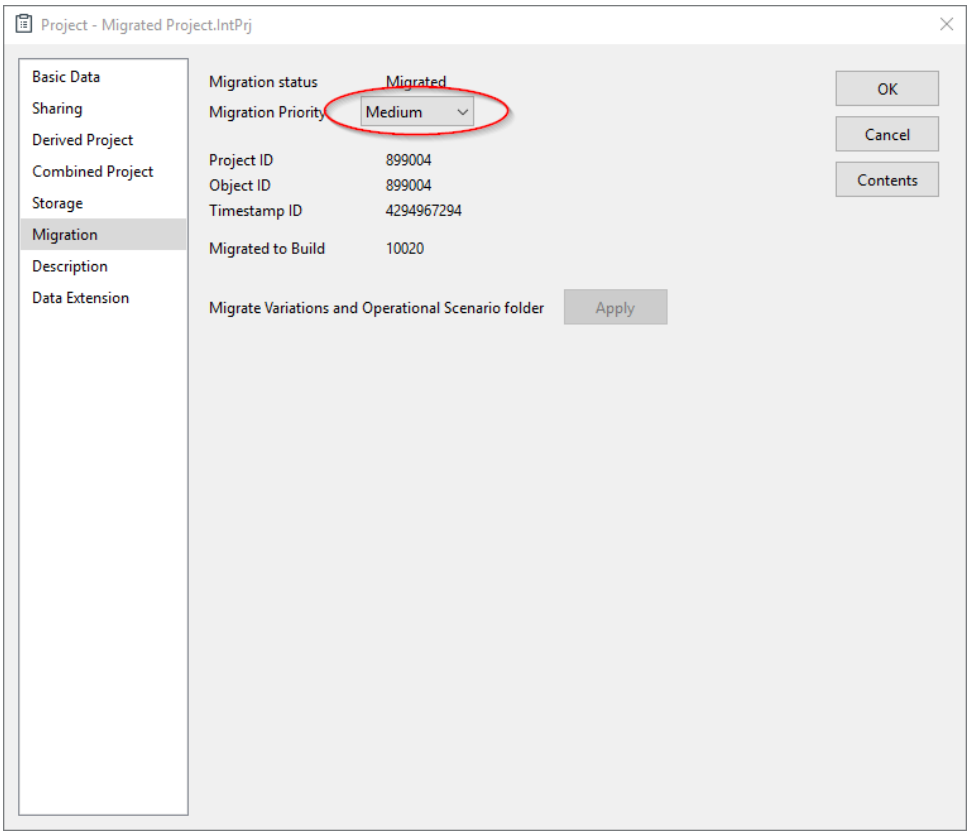

Figure 9.3.5: Migration Priority of a project

Here's a possible scenario for upgrading a big multi-user database:

1. Minimal Migration: during that time the system can not be used by any *PowerFactory* user.
2. A ***Windows*** task is created and scheduled to start *PowerFactory* in Migration Mode to run for some hours during each night. During that time users might change the migration priority to `High` of projects they think they need in the next days. After all projects have been migrated, the job can be removed.
3. Immediately after the Minimal Migration *PowerFactory* users can use the system, and migrate the projects that they're actually working on.

9.3.4 Engine Licences (GUI disabled)

Migrating a database for a *PowerFactory* installation using only engine licences requires a two step approach via the command line for a full migration.

1. Run a minimal migration to ensure the database can be read from the new version
2. Perform a project migration if the engines need to access already existing projects

A minimal migration can be started via the command line using the following command:

```
PowerFactory.exe /minimalMigration [/password:<password for Administrator>]
```

If the Administrator user in *PowerFactory* has a password set it is required to specify this password via the command line. An empty password must not be specified and the parameter can be omitted.

After the minimal migration has terminated successfully, run *PowerFactory* in *Migration Mode* as described in section 9.3.3 on page 83.

Chapter 10

Support

For additional information and help in case of problems, please visit our Support Centre at https://www.digsilent.de/en/support.html.

10.1 Support Package

Handling and analysis of support cases often requires more information about the environment, log files and various application settings. To make the collection of all these details more convenient, *PowerFactory* provides a built-in function to collect all relevant information in a so called 'Support Package'. Such a package must be created manually and should be provided when requested by the support team.

Hint: Please note: This support package contains detailed diagnostic information about the computer and its environment. These data should be checked and, if necessary, made anonymous before being passed on to third parties. This is the sole responsibility of the user.

10.1.1 Creation

A support package can easily be created from within the application via menue 'Help' → 'Support' → 'Create support package...' or via a shortcut called 'Create Support Package' in the *PowerFactory* group of the Windows start-menu.

Hint: Before creating a support package, it is strongly recommended to re-produce the issue with full logging enabled: Start *PowerFactory*, enable full logging via 'Help' → 'Support' → 'Full logging enabled' and re-produce the problem. Afterwards create a support package as described above.

10.1.2 Contents

For transparency, the package is stored as a standard zip archive and contains the following data in plain text files. It's the user's responsibility to remove all sensitive data before passing the package to someone else.

- User log files
- Application startup log files

- Crashdump files for the last 10 PowerFactory crashes
- Licence Manager log files
- Licence provider log file (CmDust.log)
- Text file listing all environment variables
- Text file listing version, installation directory, system and database timestamps, computer name, Windows user name, command line and PowerFactory.ini path used at application startup
- Text file listing all content from the installation directory
- Text file listing all ODBC drivers found
- Currently used PowerFactory configuration (PowerFactory.ini)
- Text file listing currently allocated licences
- Text file listing all found Python installations
- Text file containing a dump of the [HKEY_LOCAL_MACHINE \SOFTWARE \DIgSILENT GmbH] registry node
- Text file listing all currently running processes

ABOUT DIGSILENT

DIgSILENT was founded in 1985 and is a fully independent and privately owned company located in Gomaringen close to Stuttgart, Germany. DIgSILENT continued expansion by establishing offices in Australia, South Africa, Italy, Chile, Spain, France, the USA and Oman, thereby facilitating improved service following the world-wide increase in usage of its software products and services. DIgSILENT has established a strong partner network in many countries such as Mexico, Malaysia, UK, Switzerland, Colombia, Brazil, Peru, China and India. DIgSILENT services and software installations are used in more than 150 countries.

POWERFACTORY

DIgSILENT produces the leading integrated power system analysis software PowerFactory, which covers the full range of functionality from standard features to highly sophisticated and advanced applications including wind power, distributed generation, real-time simulation and performance monitoring for system testing and supervision. For various applications, PowerFactory has become the power industry's de-facto standard tool, due to PowerFactory models and algorithms providing unrivalled accuracy and performance.

STATIONWARE

StationWare is a central asset management system for primary and secondary equipment. In addition to handling locations and devices in a user-definable hierarchy, the system allows manufacturer-independent protection settings to be stored and managed in line with customer-specific workflows. It facilitates the management of a wide variety of business processes within a company and centralises the storage of documents. StationWare can be integrated seamlessly into an existing IT environment and the interface with PowerFactory enables the transfer of calculation-relevant data for protection studies.

MONITORING SYSTEMS

Our Power System Monitoring PFM300 product line features grid and plant supervision, fault recording, and power quality and grid characteristics analysis. The Grid Code Compliance Monitoring PFM300-GCC system also offers compliance auditing of power plants with respect to grid code requirements. This monitoring and non-compliance detection provides the complete transparency and assurance required by both plant operators and utilities.

TESTING AND CERTIFICATION

The DIN EN ISO/IEC 17025 accredited DIgSILENT Test Laboratory for NAR Conformity carries out measurements in accordance with FGW TR3 on the operational type 1 generation plant (directly coupled synchronous machines). These measurements are carried out in accordance with the "individual verification procedure" as required by the German grid connection guidelines VDE-AR-N 4110/20/30. DIgSILENT has many years of international expertise in the field of generation and consumption/load systems testing. The in-house developed and produced measuring systems enable the testing laboratory to offer customised measuring solutions for a wide range of power plants and applications.

SERVICES

DIgSILENT GmbH is staffed with experts of various disciplines relevant for performing consulting services, research activities, user training, educational programs and software development. Highly specialised expertise is available in many fields of electrical engineering applicable to liberalised power markets and to the latest developments in power generation technologies such as wind power and distributed generation. DIgSILENT has provided expert consulting services to several prominent PV and wind grid integration studies.