

# Guide to unattended installation

OWN

qualiware

## Contents

1	About This Document .....	4
2	Introduction .....	4
3	Installing QEF using unattended installer (console) .....	4
3.1	Command line parameters .....	4
3.2	New installation .....	4
3.3	Existing installation .....	5
3.4	Uninstallation.....	5
3.5	Parameters.....	5
3.6	Passwords .....	8
4	Appendix 1: The default example .....	9

## Change log

Date	Change
<b>10-11-2022</b>	Version 1 of this document. New feature in QualiWare 10.8

## 1 About This Document

QualiWare version 10.8 now includes an unattended installer. This document is intended for technical IT departments.

## 2 Introduction

The unattended installer is meant for IT departments with high volume of installations, automatic deployment environments and environments with a high focus on correct initial configuration. For single deployments, the tutorial based graphical installer is recommended.

The unattended installer will install QualiWare Execution Framework (QEF) and all available modules. After the installation configuration of AccessLog databases, Repository databases, licenses and directory services must be completed.

## 3 Installing QEF using unattended installer (console)

The unattended installer is a command line tool. This section will cover the parameters and how to execute the initial installation and how to perform upgrades.

### 3.1 Command line parameters

The console installer (SuiteInstallerC.exe) supports numerous parameters controlling different aspects of the installation. These parameters could be specified either directly as command line arguments, combined into a configuration file or both. In the latter case the ones specified as command line argument have higher priority than from the configuration file.

Overall installer recognizes the following parameter types:

- Mode: /install, /uninstall, /change, /repair
- Path to configuration file: "/DataPath:path\_to\_config\_file.cfg". Use absolute path.
- Individual parameters: "/p:ParameterName=value"

Please refer to the parameters list and brief description in section 3.5

### 3.2 New installation

To start the installation from scratch the following minimal set of parameters are required:

- Target installation folder
- QEF instance name
- QEF DB connection string
- QEF user/password on behalf of which modules will be installed and all API-based installation steps will be performed
- Modules' DB connection strings

```
Example: SuiteInstallerC.exe /install "/p:QefFolder=C:\Program Files\Qualiware\QEF"
"/p:InstanceName=QEF" "/p:QefDatabase=provider name=DatabaseProviders.Qef.MSSql;schema
name=dbo;data source=(local);initial catalog=QEF;integrated security=True"
"/p:QefUserName=Admin" "/p:QefUserPwdCmd=Admin"
"/p:QisDatabaseRepositoryDefinitionDb=provider name=DatabaseProviders.Qis.MSSql;schema
```

```
name=qis;data source=(local);initial catalog=QEF;integrated security=True"
"/p:QnsDatabaseQnsDb=provider name=DatabaseProviders.Qns.MSSql;schema name=qns;data
source=(local);initial catalog=QEF;integrated security=True"
```

In case configuration file is used: SuiteInstallerC.exe /install "/DataPath:C:\QWSetup\installData.cfg"

Alternatively, install.bat could be used. The install.bat uses pre-configured configuration file installData.cfg with all the parameters.

NOTE: installData.cfg must be edited manually beforehand in order to correctly set up minimal set of parameters (QefFolder, InstanceName, QefDatabase, QefUserName, QefUserPwd or QefUserPwdCmd, QisDatabaseRepositoryDefinitionDb or QisCmdDatabaseRepositoryDefinitionDb, QnsDatabaseQnsDb or QnsCmdDatabaseQnsDb).

### 3.3 Existing installation

To upgrade the existing installation, it is enough to specify just /DataPath parameter to the existing installation's configuration file: [InstalationPath]\Setup\installData.cfg.

Example: SuiteInstallerC.exe /change "/DataPath:C:\Program Files\QualiWare\QEF\Setup\installData.cfg"

In case of upgrading from version prior to 10.5.1 the user name and password will be needed in order to upgrade modules and repository data storages: SuiteInstallerC.exe /change "/DataPath:C:\Program Files\QualiWare\QEF\Setup\installData.cfg" "/p:QefUserName=Admin" "/p:QefUserPwdCmd=Admin"

Alternatively, change.bat with config file parameter could be used: change.bat "C:\Program Files\QualiWare\QEF\Setup\installData.cfg"

NOTE: with change.bat it is impossible to specify additional individual command line parameters.

### 3.4 Uninstallation

To uninstall existing instance from command line: SuiteInstallerC.exe /uninstall "/DataPath:C:\Program Files\QualiWare\QEF\Setup\installData.cfg"

Alternatively, using uninstall.bat: uninstall.bat "C:\Program Files\QualiWare\QEF\Setup\installData.cfg"

### 3.5 Parameters

Special command line parameters:

- **Mode:** The installer mode (install/change/repair/uninstall). Example: /install.
- **DataPath:** The absolute path to configuration file with parameters defined. Example: "/DataPath:C:\Deployment\configData.cfg"

Individual parameters ("/p:ParameterName=value")

1. **QefFolder:** The target folder (absolute path) of QualiWare platform (QEF and modules) are installed to.
2. **InstanceName:** The name of QualiWare platform instance.
3. **QefPort:** The first port from 20 ports reserved by QualiWare platform. Will be used by QEF. - 1 for automatic selection.
4. **QefSpnSkipCheck:** Whether to skip SPN check for a service account used by QEF service (usually no AD installations).
5. **QefDatabase:** The connection string to QEF database. If SQL authentication is used and plain password is specified the password will be encoded by installer and saved into installData.cfg.
6. **QefUserName:** The user name on behalf of which modules will be installed and all QEF/module-API-based steps are performed.
7. **QefUserPwdCmd:** The user password. Unencoded. Will be encoded by installer and saved into installData.cfg as QefUserPwd in encoded way for later use.
8. **QefUserPwd:** The encoded user password. Will be saved into installData.cfg as is for later use.
9. **QisCmdDatabaseRepositoryDefinitionDb:** The connection string to QIS module database. If SQL authentication is used and plain password is specified the password will be encoded by installer and saved into installData.cfg as QisDatabaseRepositoryDefinitionDb for later use.
10. **QisDatabaseRepositoryDefinitionDb:** The connection string to QIS module database. If SQL authentication is used and plain password is specified the password will be encoded by installer and saved into installData.cfg for later use.
11. **QnsCmdDatabaseQnsDb:** The connection string to QNS module database. If SQL authentication is used and plain password is specified the password will be encoded by installer and saved into installData.cfg as QnsDatabaseQnsDb for later use.
12. **FullInstanceName:** Usually equals to InstanceName.
13. **WindowsFeaturesSkipCheck:** Whether to skip checking windows features prerequisites.
14. **QefBinFolder:** Usually calculated automatically from QefFolder.
15. **QefBinFolder:** Usually calculated automatically from QefFolder.
16. **QefSetupFolder:** Usually calculated automatically from QefFolder.
17. **QefWebSite:** The number of root IIS site to use. Usually defaults to 1 (Default Web Site).
18. **QefWebApplicationPool:** The IIS application pool to use. Defaults to DefaultAppPool.
19. **QefWebAuthenticationType:** The authentication type in IIS for QualiWare sites and resources. Defaults to Anonymous. Authentication of users is performed by QEF.
20. **QefIisPath:** IIS path to root QEF web application. Defaults to IIS://localhost/W3SVC/1/ROOT/QEF.
21. **QefWebPhysicalPath:** Path to root QEF web folder on a disk. Usually calculated automatically as QefFolder\Web.
22. **QefWebPathType:** Defaults to Qef.
23. **QefWebVirtualDirectory:** Defaults to QEF.
24. **QefConsoleFolder:** Path to QEF Console on a disk. Usually calculated automatically as QefWebPhysicalPath\Console.
25. **QefHttpAccessBinding:** The http binding to listen on. Read from IIS by default. Usually \*:80:.
26. **QefHttpAddress:** The root URI of QEF http address. Defaults to http://localhost/.
27. **QefConsoleHttpUrl:** The URL of QEF Admin Console. Defaults to http://localhost/QEF/Console.

28. **QefHttpAccessEnabled:** Whether to enable http access to QEF and listen for http requests.
29. **QefHttpsAccessBinding:** The https (SSL) binding to listen on. Read from IIS by default. Usually \*:443:.
30. **QefHttpsAddress:** The root URI of QEF https (SSL) address. Defaults to https://localhost/.
31. **QefConsoleHttpsUrl:** The URL of QEF Admin Console using SSL. Defaults to https://localhost/QEF/Console.
32. **QefHttpsAccessEnabled:** Whether to enable https (SSL) access to QEF and listen for https requests.
33. **QefShortcutsDesktop:** Whether to create shortcuts on desktop.
34. **QefShortcutsStartMenu:** Whether to create shortcuts in start menu.
35. **QefShortcutsAllUsers:** Whether to create shortcuts for all users or current user only.
36. **TcpPortRanges:** Defines variables for port ranges used by instance. Defaults to QefModulesPortRangeStart,QefModulesPortRangeEnd.
37. **QefModulesPortRangeStart:** The first port from range used by QEF instance.
38. **QefModulesPortRangeEnd:** The last port from range used by QEF instance.
39. **QefPortRangeFirewallRuleName:** The name for windows firewall rule allowing the port range for QEF instance.
40. **TcpPorts:** Defaults to QefWcfPort.
41. **QefWcfPort:** First port used by WCF.
42. **QefWcfFirewallRuleName:** The name for windows firewall rule allowing the WCF port for QEF instance.
43. **QefServicePreservePassword:** Whether to preserve password for service account QEF is run under.
44. **QefServiceSkipRegistration:** Whether to skip registering QEF as a windows service.
45. **QefServiceName:** Name of windows service for QEF.
46. **QefServiceDisplayName:** The user-friendly display name of QEF windows service.
47. **QefServiceStartupType:** QEF windows service startup type.
48. **QefServiceUser:** The service account/user name for QEF windows service.
49. **QefServicePassword:** The password of QEF windows service account/user.
50. **HttpListenAccessSkipReservation:** Whether to skip reservation of http port listen access.
51. **QefWebPhysicalPathSkipAccessCheck:** Whether to skip checking physical access to QefWebPhysicalPath.
52. **QefModulesFolder:** The modules' folder name. Defaults to Modules.
53. **QefDatabaseSkipInstallAccessCheck:** Whether to skip checking access rights to QEF database on behalf of credentials at installation time.
54. **QefDatabaseServerAccess:** Calculated. Usually DBInstallerCredentialsAccess.
55. **QefDatabaseCanBeCreated:** Whether database can be created by installer.
56. **QefDatabaseExists:** Calculated. Whether defined QEF database exists or not.
57. **QefDataStorageExists:** Calculated. Whether defined data storage within the database exists or not.
58. **QefDatabaseSkipCreation:** Whether to skip QEF database creation from installer.
59. **QefDatabaseSkipRuntimeAccessCheck:** Whether to skip checking access rights to QEF database on behalf of credentials defined for QEF runtime use (QefDatabase).
60. **QefDatabaseRuntimeAccess:** Calculated. Whether QEF has access to its database.
61. **LicensesFolder:** Absolute path to folder to install licenses from during installation.
62. **QtsDeploymentPackageName:** The name of QualiWare Task Scheduler (QTS) module bundle package name. Defaults to QtsBundle.zip.

63. **QdcDeploymentPackageName:** The name of QualiWare Connector for SharePoint Server (QDC) module bundle package name. Defaults to QdcBundle.zip.
64. **QisDeploymentPackageName:** The name of QualiWare Integration Server (QIS) module bundle package name. Defaults to QisBundle.zip.
65. **QlmDeploymentPackageName:** The name of QualiWare Lifecycle Manager (QLM) module package name. Defaults to QlmModule.zip.
66. **QnsDeploymentPackageName:** The name of QualiWare Notification Service (QNS) module package name. Defaults to Qns.zip.
67. **QlmDocDeploymentPackageName:** The name of QualiWare Lifecycle Manager Documentation module package name. Defaults to QlmDocumentationModule.zip.
68. **QisQIS\_AdministratorRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QIS role QIS Administrator.
69. **QnsQNS\_AdministratorRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QNS role QNS Administrator.
70. **QlmQLM\_AdministratorRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QLM role QLM Administrator.
71. **QtsQTS\_AdministratorRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QLM role QTS Administrator.
72. **QtsQTS\_Power\_UserRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QLM role QTS Power User.
73. **QtsQTS\_UserRoleAssignTo:** The list of users/groups defined by ID or login/name and delimited by ;. During installation these users and groups will be assigned to QLM role QTS User.
74. **QlmCreateSharedFolder:** Whether to create a network shared folder for QLM module (to be run from share). Defaults to False.
75. **QlmSharedFolderName:** The name of QLM network shared folder. Defaults to QLM.
76. **QlmSharedFolderDescription:** The optional description of QLM network share.
77. **HttpListenAccessPrefixes:** Prefixes to differentiate QEF http listener. Defaults to empty.
78. **QefPredefinedSettings:** Defaults to empty.
79. **QefCustomSettings:** Defaults to empty.
80. **QefConsolePredefinedSettings:** Defaults to empty.
81. **QefConsoleCustomSettings:** Defaults to empty.

## 3.6 Passwords

Passwords can be added with or without encryption. Without leave a significant opening in security. Here follows a guide how to encoded password.

1. Extract any .zip module file - QCommon.dll is needed from there.
2. Execute the below powershell commands to encode password (from the folder where QCommon.dll is located):
3. Add-Type -Path "QCommon.dll"
4. [QCommon.Extensions.StringExtensions]::Encrypt("MyPlainStringPassword")

This will output the encoded password value.

## 4 Appendix 1: The default example

This example is as well included in the installation package

```
<?xml version="1.0" encoding="utf-8"?>
<install>
  <data name="QefFolder">C:\Program Files\Qualiware\QEF</data>
  <data name="InstanceName">QEF</data>
  <data name="QefPort">-1</data>
  <data name="QefSpnSkipCheck">True</data>
  <data name="QefDatabase">provider name=DatabaseProviders.Qef.MSSql;schema name=dbo;data
source=(local);initial catalog=QEF;integrated security=True</data>
  <data name="QefUserName">Admin</data>
  <!-- Encoded password. To pass plain value (will be encoded and saved) use QefUserPwdCmd -->
  <data
name="QefUserPwd">0Djy/J6QP/xHsiAAI7pgd3mP/97bX2XbPG4q3RuDYnZLwrOpOU2yydM+wi3Yq3
8M</data>
  <!-- Encoded connection string. To pass plain value (will be encoded and saved) use
QisCmdDatabaseRepositoryDefinitionDb -->
  <data name="QisDatabaseRepositoryDefinitionDb">provider
name=DatabaseProviders.Qis.MSSql;schema name=qds_dbo;data source=(local);initial
catalog=QEF;integrated security=True</data>
  <!-- Encoded connection string. To pass plain value (will be encoded and saved) use
QnsCmdDatabaseQnsDb -->
  <data name="QnsDatabaseQnsDb">provider name=DatabaseProviders.Qns.MSSql;schema
name=qns;data source=(local);initial catalog=QEF;integrated security=True</data>
  <!--
  <data name="FullInstanceName">QEF</data>
  <data name="WindowsFeaturesSkipCheck">False</data>
  <data name="QefBinFolder">C:\Program Files\Qualiware\QEF\Bin</data>
  <data name="QefExecutable">C:\Program Files\Qualiware\QEF\Bin\Qef.exe</data>
  <data name="QefSetupFolder">C:\Program Files\Qualiware\QEF\Setup</data>
  <data name="QefWebSite">1</data>
  <data name="QefWebApplicationPool">DefaultAppPool</data>
  <data name="QefWebAuthenticationType">Anonymous</data>
```

```

<data name="QefIisPath">IIS://localhost/W3SVC/1/ROOT/QEF</data>
<data name="QefWebPhysicalPath">C:\Program Files\Qualiware\QEF\Web</data>
<data name="QefWebPathType">Qef</data>
<data name="QefWebVirtualDirectory">QEF</data>
<data name="QefConsoleFolder">C:\Program Files\Qualiware\QEF\Web\Console</data>
<data name="QefHttpAccessBinding">*:80:</data>
<data name="QefHttpAddress">http://localhost/</data>
<data name="QefConsoleHttpUrl">http://localhost/QEF/Console</data>
<data name="QefHttpAccessEnabled">True</data>
<data name="QefHttpsAccessBinding">*:443:</data>
<data name="QefHttpsAddress">https://localhost/</data>
<data name="QefConsoleHttpsUrl">https://localhost/QEF/Console</data>
<data name="QefHttpsAccessEnabled">True</data>
<data name="QefShortcutsDesktop">True</data>
<data name="QefShortcutsStartMenu">True</data>
<data name="QefShortcutsAllUsers">True</data>
<data name="TcpPortRanges">
  <i00>QefModulesPortRangeStart,QefModulesPortRangeEnd</i00>
</data>
<data name="QefModulesPortRangeStart">25780</data>
<data name="QefModulesPortRangeEnd">25799</data>
<data name="QefPortRangeFirewallRuleName">QEF Port Range</data>
<data name="TcpPorts">
  <i00>QefWcfPort</i00>
</data>
<data name="QefWcfPort">25780</data>
<data name="QefWcfFirewallRuleName">QEF WCF</data>
<data name="QefServicePreservePassword">False</data>
<data name="QefServiceSkipRegistration">False</data>
<data name="QefServiceName">QEF</data>
<data name="QefServiceDisplayName">QEF</data>

```

```

<data name="QefServiceStartupType">Automatic</data>
<data name="QefServiceUser">
  <Domain></Domain>
  <UserName>SYSTEM</UserName>
  -->
  <!-- HOSTNAME should be replaced with the real name of a host machine -->
  <!--
  <SamUserName>HOSTNAME\SYSTEM</SamUserName>
  <IsLocal>True</IsLocal>
  <IsMsa>False</IsMsa>
</data>
-->
<!-- Password value must be encoded. -->
<!--
<data name="QefServicePassword"></data>
<data name="HttpListenAccessSkipReservation">False</data>
<data name="QefWebPhysicalPathSkipAccessCheck">False</data>
<data name="QefModulesFolder">Modules</data>
<data name="QefDatabaseSkipInstallAccessCheck">False</data>
<data name="QefDatabaseServerAccess">DBInstallerCredentialsAccess</data>
<data name="QefDatabaseCanBeCreated">True</data>
<data name="QefDatabaseExists">True</data>
<data name="QefDataStorageExists">True</data>
<data name="QefDatabaseSkipCreation">False</data>
<data name="QefDatabaseSkipRuntimeAccessCheck">False</data>
<data name="QefDatabaseRuntimeAccess">True</data>
-->
<!-- Path to folder to install licenses from -->
<!--
<data name="LicensesFolder">C:\Licenses</data>
<data name="QtsDeploymentPackageName">QtsBundle.zip</data>

```

```

<data name="QdcDeploymentPackageName">QdcBundle.zip</data>
<data name="QisDeploymentPackageName">QisBundle.zip</data>
<data name="QImDeploymentPackageName">QImModule.zip</data>
<data name="QnsDeploymentPackageName">Qns.zip</data>
<data name="QImDocDeploymentPackageName">QImDocumentationModule.zip</data>
-->
<!-- Assign module roles to user(s) and/or group(s), joined by ; -->
<!--
<data name="QisQIS_AdministratorRoleAssignTo">Admin</data>
<data name="QnsQNS_AdministratorRoleAssignTo">Admin</data>
<data name="QImQLM_AdministratorRoleAssignTo">Admin</data>
<data name="QtsQTS_AdministratorRoleAssignTo">Admin</data>
-->
<!-- QLM network share -->
<!--
<data name="QImCreateSharedFolder">False</data>
<data name="QImSharedFolderName">QLM</data>
<data name="QImSharedFolderDescription">QLM share</data>

<data name="HttpListenAccessPrefixes" />
<data name="QefPredefinedSettings" />
<data name="QefCustomSettings" />
<data name="QefConsolePredefinedSettings" />
<data name="QefConsoleCustomSettings" />
<groups>
  <group name="KnownFolders">
    <var name="QefBinFolder" />
    <var name="QefSetupFolder" />
    <var name="QefWebPhysicalPath" />
    <var name="QefModulesFolder" />
  </group>

```

```
<group name="WebAccessEnabledFlags">
  <var name="QefHttpAccessEnabled" />
  <var name="QefHttpsAccessEnabled" />
</group>
<group name="InstallerControlledSettings">
  <var name="QefPortRangeStart" />
  <var name="QefPortRangeEnd" />
  <var name="port-wcf" />
  <var name="ConnectionString" />
  <var name="InstanceName" />
  <var name="local-wcf-netpipe" />
  <var name="QefIisPath" />
  <var name="ServiceName" />
</group>
</groups>
-->
</install>
..
```