PDC FRAMEWORK SCHEDULER

INSTALLATION GUIDE

Applications Install and Configuration Procedure

## Document History

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Comments** |
| **1.0** | 17.03.2015 |  |
| **1.1** | 10.04.2015 | comments incorporated |
| **1.2** | 11.11.2016 |  |

## Document Approval

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Title** | **Signature** | **Date** |
| **Created By** | Petr Stefanek | Sr. Solution Architect |  | 17.03.2015 |
| **Verified** |  |  |  |  |
| **Approved** |  |  |  |  |

## Referred Documents and Forms

|  |  |  |
| --- | --- | --- |
| **No** | **Code** | **Title** |
|  |  |  |

Contents

[Document History 2](#_Toc466847141)

[Document Approval 2](#_Toc466847142)

[Referred Documents and Forms 2](#_Toc466847143)

[Contents 3](#_Toc466847144)

[1. PDC Generalities 6](#_Toc466847145)

[2. PDC Installation Overview 7](#_Toc466847146)

[2.1 Permissions and privileges 7](#_Toc466847147)

[2.2 Minimum hardware requirements 7](#_Toc466847148)

[2.3 Software requirements 7](#_Toc466847149)

[2.4 Port Opening 8](#_Toc466847150)

[3. Step by step installation 9](#_Toc466847151)

[3.1 PDC repository creation 9](#_Toc466847152)

[3.1.1 NLS\_LANG settings 9](#_Toc466847153)

[3.1.2 Sqlnet.ora settings 9](#_Toc466847154)

[*3.1.3* *tnsnames.ora* 9](#_Toc466847155)

[3.1.4 Control metadata population 9](#_Toc466847156)

[3.1.5 Changes in CTRL\_PARAMETERS table 10](#_Toc466847157)

[3.1.6 Metadata Setup note 10](#_Toc466847158)

[3.1.7 SQL code for table content correctness checking 10](#_Toc466847159)

[3.2 System Environment Variables Setup 12](#_Toc466847160)

[3.3 Server directory setup 13](#_Toc466847161)

[3.3.1 Security settings on Security directory 13](#_Toc466847162)

[3.3.2 System\_info.xml file populating 13](#_Toc466847163)

[3.4 Engine Setup 13](#_Toc466847164)

[3.4.1 Necessary changes in Engine.pl script 14](#_Toc466847165)

[3.5 Framework\_checker Setup 15](#_Toc466847166)

[3.6 Initialization Setup 15](#_Toc466847167)

[3.7 PDC GUI Installation and Setup 16](#_Toc466847168)

[3.7.1 WEB installation and configuration 16](#_Toc466847169)

[3.7.1.1 Check application tools settings before creation of website/application 16](#_Toc466847170)

[3.7.1.2 Add PDC instance as an application of the website 17](#_Toc466847171)

[3.7.1.3 Add PDC instance as new website 18](#_Toc466847172)

[3.7.2 Configuration file: 19](#_Toc466847173)

[3.7.3 Security settings on D:\Inetpub\PDCxxx\Web.config file 19](#_Toc466847174)

[3.7.4 GUI Access permissions management 20](#_Toc466847175)

[3.7.5 User Authentication 20](#_Toc466847176)

[3.7.6 Page Authorization 20](#_Toc466847177)

[3.7.7 User Management Support 21](#_Toc466847178)

[3.7.8 Access Control 22](#_Toc466847179)

[3.7.9 Password Change 22](#_Toc466847180)

[3.8 Task scheduling 23](#_Toc466847181)

[3.8.1 Engine 24](#_Toc466847182)

[3.8.2 Initialization 27](#_Toc466847183)

[3.8.3 Framework checker 30](#_Toc466847184)

[4. Environment preparation 32](#_Toc466847185)

[4.1 ActiveState PERL Installation (licensed) 32](#_Toc466847186)

[4.2 Oracle Database Installation and Setup (licensed) 35](#_Toc466847187)

[4.3 Oracle Client SW Installation 36](#_Toc466847188)

[4.4 .Net Framework Installation 36](#_Toc466847189)

[4.5 Microsoft IIS Installation (licensed) 36](#_Toc466847190)

# PDC Generalities

**Process Dispatch Center** (PDC) is a controlling framework application, which is used for optimization of ETL/ELT process and its supporting tasks. The main task of PDC is launching separate jobs in sequence based on their dependencies and priorities. While doing so it utilizes optimally systems resources and uses parallelism as much as possible. The frameworks logic takes care about optimal launch of all jobs, while the internal logic of the job itself is invisible to the framework. Framework runs all jobs in the same manner – using a child process, which interprets the content of the job command line – and waits for the job exit code. In dependency on the exit code the framework considers the job finished either successfully or failed. All necessary information needed by the framework is stored as metadata in Oracle database.

# PDC Installation Overview

The objective of this PDC installation guide is to make you familiar with installation process for this solution. In practice, PDC solution depends on several parties, which have to be installed first. Some of them are licensed such Oracle database, ActiveState PERL and Microsoft IIS. Each part of PDC framework can be installed on separate HW or all parties can be install on common HW.

## Permissions and privileges

For PDC installation you must have the following permissions and privileges:

1. For installation purpose you have to have administrative on target machine
2. Administrative privilege is also necessary for component settings such as environment variable settings, for directory guarding, connectivity settings, Microsoft IIS configuration and so on.
3. Oracle administrative right (sysdba) are also proffered for Oracle objects creation, but administrative rights not necessary for regular execution, right can be reduced after installation

## Minimum hardware requirements

|  |  |
| --- | --- |
| **Component** | **Storage** |
| Oracle database | 10+ GB depends on history length keeping |
| ActiveState PERL | 200 MB |
| Microsoft IIS | 2 MB |

## Software requirements

|  |  |  |
| --- | --- | --- |
| **Type** | **Minimal** | **Optional** |
| Operating System  Engine, Checker | MS Windows Server 2003 | MS Windows Server 2003 and above |
| PERL ActiveState | ActiveState PERL 5.14 for used OS | ActiveState PERL 5.16 or above for used OS |
| Web browser | Microsoft Internet Explorer 8.x | Microsoft Internet Explorer 8.x and above |
| Oracle | Oracle 11g Express edition | Oracle 11g |
| Oracle Client | Oracle Database 11g Release 2 Client | Oracle Database 11g Release 2 Client |
| .Net framework | .Net framework 4.0 | .Net framework 4.5 or above |

## Port Opening

Preferred way is not block any communication between PDC framework members. In case when security doesn’t allow port opening, give the pass at least communication on ports mentioned in table bellow.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Source** | **Target** | **Protocol** | **Port** | **Service** | **Description** |
|  | Engine | TCP  TCP, UDP  TCP, UDP | 3389  445  135 | RDP  Netbios over TCP/IP  Windows RPC | Remote desktop  Windows share |
| Engine | Oracle server | TCP | 1521 or other | Oracle | Oracle listener |
| PCs | GUI | TCP | 80 or other | HTTP | WEB |

# Step by step installation

At this point we suppose that all SW components are already properly installed and configured:

* ActiveState PERL
* Oracle Database
* Oracle client
* .Net Framework
* Microsoft IIS

Installation and configuration details are discussed in chapter 4 - “Environment preparation” in this document.

## PDC repository creation

PDC repository content are delivered as Oracle database export in six files named:

* 000\_droping\_old\_structures.sql
* 001\_Object\_structure-PDC\_repository.sql
* 002\_Package-PDC\_repository.sql
* 003\_Trigger-PDC\_repository.sql
* 004\_Object\_data-PDC\_repository.sql
* 005\_Engine0\_creation-PDC\_repository.sql

File application is provided in numbered order.

Use **sqlplus** command for connecting Oracle database as pdc user. Run enclosed files in numbered order. After application each file check application errors and fix it before applying next file. Mostly the errors are caused due insufficient database rights.

File 000\_droping\_old\_structures.sql contains DROP object command on beginning for easy repeating of file content application. Please note, that these commands violate errors generation in initial run.

### NLS\_LANG settings

Use regedit command to check if NLS\_LANG is set to value AMERICAN\_AMERICA.WE8MSWIN1252 in HKLM\Software\Oracle

### Sqlnet.ora settings

N*o special requirements for settings, user standard (default) one.*

### *tnsnames*.ora

Define TNS connection used for connection to PDC repository.

### Control metadata population

**Supplied files contain only technology streams and jobs. You need insert own metadata into control (CTRL) tables for PDC functionality:**

* CTRL\_STREAM – stream information
* CTRL\_STREAM\_DEPENDENCY – inter stream dependency in format child-parent relation
* CTRL\_JOB – list of all jobs and its run parameters, job to stream relation
* CTRL\_JOB\_DEPENDENCY – inter job dependency inside stream, can be used also for job to job relation belongs to different stream which is not recommended
* CTRL\_STREAM\_PLAN\_REF – calendar for job launch planning
* CTRL\_JOB\_TABLE\_REF – which table is written or red by a job

### Changes in CTRL\_PARAMETERS table

Set correct information in CTRL\_PARAMETERS table. Change values in these param\_name parameters:

* ENGINE\_STATUS
* ENVIRONMENT
* SYSTEM\_OFF

### Metadata Setup note

*In future all necessary metadata changes in control tables will be mentioned here. Please note expanded structure of CTRL\_JOB table where column called “toughness” \*) has been added. Adequate metadata for this new functionality support are stored in CTRL\_TASK\_PARAMETERS table.*

*\*) Toughness defines how many resources are consumed by job. For example if weight of SMALL job is 10 and weight of MEDIUM job is 20, you can run 10 SMALL jobs or 5 MEDIUM jobs in category throughput equal 100. Standardly we support four toughness:*

* *SMALL*
* *MEDIUM*
* *LARGE*
* *EXCEPTIONAL*

*All toughness weights of every job category are stored in CTRL\_TASK\_PARAMETERS (see metadata examples stored in table). If no predefined toughness is valid for specific job, you can define job’s exception in the same table.*

Finally you have to populate also lookup (LKP) tables and CTRL\_TASK\_PARAMETERS table. Use SQL code for table correctness checking.

### SQL code for table content correctness checking

Check if CTRL tables contain some data:

*SELECT COUNT(\*) FROM CTRL\_JOB;*

*/\* Rows have to be returned \*/*

*SELECT COUNT(\*) FROM CTRL\_JOB\_DEPENDENCY;*

*/\* Rows have to be returned \*/*

*SELECT COUNT(\*) FROM CTRL\_STREAM;*

*/\* Rows have to be returned \*/*

*SELECT COUNT(\*) FROM CTRL\_STREAM\_DEPENDENCY;*

*/\* Rows have to be returned \*/*

*SELECT COUNT(\*) FROM CTRL\_STREAM\_PLAN\_REF;*

*/\* Rows have to be returned \*/*

Check job duplicity in CTRL\_JOB table:

*SELECT job\_name, COUNT(\*)*

*FROM CTRL\_JOB*

*GROUP BY job\_name*

*HAVING COUNT(\*) > 1;*

*/\* No rows should be returned \*/*

Looking for unsigned job to stream in CTRL\_JOB table:

*SELECT \* from ctrl\_job*

*where stream\_name not in (*

*select stream\_name*

*from ctrl\_stream);*

*/\* No rows should be returned \*/*

Looking for stream having no calendar in CTRL\_STREAM\_PLAN\_REF table:

*SELECT \* FROM CTRL\_JOB*

*WHERE stream\_name NOT IN (*

*SELECT stream\_name*

*from ctrl\_stream\_plan\_ref);*

*/\* No rows should be returned \*/*

Looking for dependency between non existing jobs:

*SELECT \* FROM CTRL\_JOB*

*select \* from CTRL\_JOB\_DEPENDENCY*

*WHERE job\_name not in (*

*SELECT job\_name*

*FROM CTRL\_JOB)*

*OR parent\_job\_name not in (*

*SELECT job\_name*

*FROM CTRL\_JOB);*

*/\* No rows should be returned \*/*

Looking for dependency between non existing streams:

*select \* from CTRL\_STREAM\_DEPENDENCY*

*WHERE stream\_name not in (*

*SELECT stream\_name*

*FROM CTRL\_STREAM)*

*OR parent\_stream\_name not in (*

*SELECT stream\_name*

*FROM CTRL\_STREAM);*

*/\* No rows should be returned \*/*

Lookup tables correctness checking:

*select \* from ctrl\_job*

*where job\_category not in (*

*select job\_category from lkp\_job\_category);*

*/\* No rows should be returned \*/*

*select \* from ctrl\_job*

*where job\_type not in (*

*select job\_type from lkp\_job\_type);*

*/\* No rows should be returned \*/*

*select \* from ctrl\_job*

*where phase not in (*

*select job\_phase from lkp\_phase);*

*/\* No rows should be returned \*/*

Looking for job’s category which is not declared in CTRL\_TASK\_PARAMETERS:

*SELECT \* FROM CTRL\_JOB*

*WHERE (job\_category, engine\_id) not in (*

*SELECT task\_subtype, engine\_id*

*FROM CTRL\_TASK\_PARAMETERS*

*WHERE param\_type = 'PARALLELISM\_CONTROL');*

*/\* No rows should be returned \*/*

*SELECT \* FROM CTRL\_JOB*

*WHERE (job\_category, engine\_id) not in (*

*SELECT task\_subtype, engine\_id*

*FROM CTRL\_TASK\_PARAMETERS*

*WHERE param\_type = 'TASK\_MIN\_CONTROL');*

*/\* No rows should be returned \*/*

Checking if CTRL\_PARAMETERS table is filled:

*SELECT \* FROM CTRL\_PARAMETERS*

*ORDER BY param\_name, param\_cd;*

*/\* Rows have to be returned \*/*

Basic correctness check of CTRL\_PARAMETERS table:

*select param\_name, count(\*)*

*from CTRL\_PARAMETERS*

*GROUP BY param\_name*

*ORDER BY param\_name;*

*/\* All numbers should dave the same number, ENGINE\_STATUS & SYSTEM\_OFF are doubled, tripled ... (depends of number of ETL servers) \*/*

## System Environment Variables Setup

System environment variables are used for script behavior setup. The main advantage of using system variables is possibility having same version of script in different environment. Use “Environment variable setting tab” on Windows operating system for environment variable settings.

Follow system environments variables are used by PDC scripts (change sample values regards your settings):

**SMTP\_DEST**=m-out.cz.telefonica

**SNMP\_DEST**=ntgdwm01

**PMROOTDIR**=F:\PDC

**PERLEXE**=C:\Perl518\bin\perl.exe

**SYSTEMNAME**=ntetl404

**PMWORKFLOWLOGDIR**=F:\PDC\Logs

**PMWORKFLOWOLDLOGDIR**=F:\PDC\OldLogs

## Server directory setup

PDC expects Engine, Initialization and Framework Checker scripts located in BIN subdirectory of %PMRootDir% directory. Unzip content of **PDC\_directory\_structure.zip** file and move content of PDC folder into your %PMRotDir% ($PMRootDir) directory.

### Security settings on Security directory

On Windows operating system set security on %PMRootDir%\Security directory using Windows explorer. Select Directory and go thru:

Properties -> Security -> Full Control ... only for defined „secure“ users. Other „unsecure“ users without any permission !

### System\_info.xml file populating

Change content of **system\_info.xml** file to be in matter of fact.

## Engine Setup

Engine.pl script located in %PMRootDir%\Bin\Framework directory has to be edited for correct interpretation of cmd\_line value. Check if all job\_type values are mentioned in $job\_type comparing and correct job’s command line are compiled from cmd\_line content. If not, make necessary adjustment. Please note that cmd\_line content has to be environmentally independent, it means that you use same cmd\_line content for starting job in development, test and production environment. This is necessary for change management process. For command line distinction in each environment, use system variables or values in system\_info.xml file instead of constant.

%PERLEXE% -w %PMRootDir%\Bin\Framework\Engine.pl -eng 0 -x 8

Used parameters:

* -eng – Emgine number
* -x – Debug level (0 = less, 9 = more). Debug level 9 shows debug information include used passwords. Due this fact its use is blocked inside script (debug level 9 is automatically decreased to debug level 8). If you need information about sent password also, unblock this future in script
* -debug – force request debug information of Oracle procedures. Do not use this option if it is not absolutely necessary, if procedure fails, it switches into debug mode automatically

### Necessary changes in Engine.pl script

You need to describe launch method of all job\_type used in control metadata definition (LKP\_JOB\_TYPE table) by editing code in section ($availability == 1, 2 and 3). (Section example)

*if ($availability == 1 or $availability == 2 or $availability == 3) {*

*if (uc($job\_type) eq "DATASTAGE" or uc($job\_type) eq "BTEQ" or uc($job\_type) eq "DS\_PERL") {*

*$cmd = $cmd\_line;*

*$cmd = $cmd . "\n";*

*}*

*elsif (uc($job\_type) eq "ABORT\_INFORMATICA") {*

*$cmd = "$ETLABORT " . $cmd\_line . "\n";*

*}*

*elsif (uc($job\_type) eq "DATA\_QUALITY" or uc($job\_type) eq "MAN\_DATA\_QUALITY") {*

*$cmd = "$cmd\_line -name $job\_name\n";*

*}*

*elsif (uc($job\_type) eq "CHECKER"*

*or uc($job\_type) eq "DELIVERY\_CHECKER"*

*or uc($job\_type) eq "EXPORT\_TGT"*

*or uc($job\_type) eq "EXPORT\_WRK"*

*or uc($job\_type) eq "HISTORIZATION"*

*or uc($job\_type) eq "LOADER\_DM"*

*or uc($job\_type) eq "LOADER\_ERR"*

*or uc($job\_type) eq "LOADER\_STG"*

*or uc($job\_type) eq "LOADER\_TGT"*

*or uc($job\_type) eq "LOADER\_WRK"*

*or uc($job\_type) eq "SNIFFER"*

*or uc($job\_type) eq "TRANSFORMATION"*

*or uc($job\_type) eq "UNIFICATION"*

*or uc($job\_type) eq "VALIDATOR") {*

*if ($availability == 1) { # START*

*$cmd = "$ETL\_START " . $cmd\_line . "\n";*

*}*

*elsif ($availability == 2) { # RESUME*

*$cmd = "$ETL\_RESUME " . $cmd\_line . "\n";*

*}*

*elsif ($availability == 3) { # RESTART*

*$cmd = "$ETL\_ABORT " . $cmd\_line . "\n";*

*$cmd = $cmd . "$ETL\_START " . $cmd\_line . "\n";*

*}*

*}*

*elsif (uc($job\_type) eq "COMMAND\_ARG") {*

*$cmd = $cmd\_line . " -availability $availability";*

*$cmd = $cmd . " -job\_id $job\_id";*

*$cmd = $cmd . " -job\_name $job\_name";*

*$cmd = $cmd . " -job\_type $job\_type";*

*$cmd = $cmd . " -job\_category $job\_category";*

*$cmd = $cmd . " -load\_date $load\_date";*

*$cmd = $cmd . " -queue\_number $queue\_number ";*

*}*

*elsif (uc($job\_type) eq "RUN\_SCRIPT") {*

*$cmd = $ENV{'PERLEXE'} . ' ' . File::Spec->catfile("$ENV{'PMRootDir'}", "Cmd", "run\_script.pl") . " $ENV{'PMRootDir'} $ENV{'PMWorkflowLogDir'} " . $cmd\_line;*

*}*

*else {*

*$cmd = $cmd\_line;*

*}*

*}*

## Framework\_checker Setup

Framework\_checker script should be use only once per environment, it checks all Engines and also all Initialization processes. Please note that in O2 environment is Framework checker installed on ntgdwm01.

Additional environment variables are set directly in Framework.bat.

%PERLEXE% -w %PMRootDir%\Bin\Framework\Framework\_checker.pl -x 8

cd %PMRootDir%\Logs\BinLogs

for /F %%A in ("Framework\_checker.log") do If %%~zA gtr 5000000 rename Framework\_checker.log Framework\_checker.log.%date:~10,4%%date:~7,2%%date:~4,2%\_%time:~0,2%%time:~3,2%

Used parameters:

* -x – Debug level (0 = less, 9 = more). Debug level 9 shows debug information include used passwords. Due this fact its use is blocked inside script (debug level 9 is automatically decreased to debug level 8). If you need information about sent password also, unblock this future in script
* -debug – force request debug information of Oracle procedures. Do not use this option if it is not absolutely necessary, if procedure fails, it switches into debug mode automatically

## Initialization Setup

Initialization initializes each Engine. Script contains environment variable settings and executable code call.

Additional environment variables are set directly in Initialization.bat.

%PERLEXE% -w %PMRootDir%\Bin\Framework\Engine.pl -eng 0 –x 8

Used parameters:

* -eng – Emgine number
* -x – Debug level (0 = less, 9 = more). Debug level 9 shows debug information include used passwords. Due this fact its use is blocked inside script (debug level 9 is automatically decreased to debug level 8). If you need information about sent password also, unblock this future in script
* -debug – force request debug information of Oracle procedures. Do not use this option if it is not absolutely necessary, if procedure fails, it switches into debug mode automatically

## PDC GUI Installation and Setup

File **PDC\_GUI.zip** contains all necessary files for PDC GUI installation. Unzip PDC\_GUI.zip file and move all files from GUI directory into C:\Inetpub\wwwroot directory. Make necessary changes in Web.config file. Database connection is stored in the ASP .NET configuration file – Web.Config. The connection in ConnectionString property is used throughout the project.

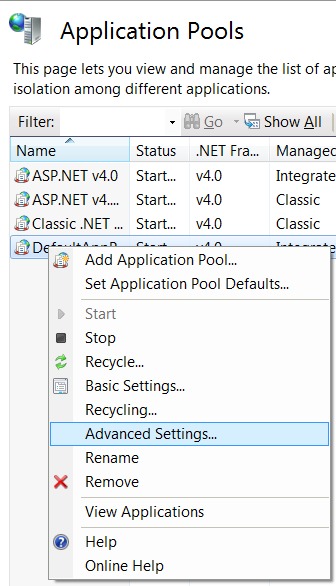
Configuration must exist for each PDC server (PDCprod, PDCtest, PDCdev).

### WEB installation and configuration

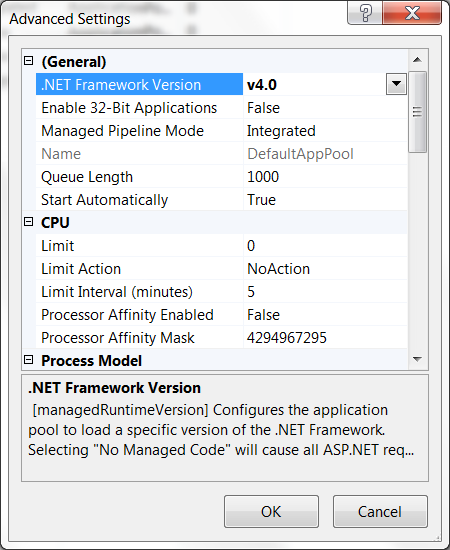
* Suppose that IIS is correctly installed and configured. See details in chapter Environment preparartion
* For all WEB create directory in D:\Inetpub\wwwroot (PDCprod, PDCdev, PDCtest).
* Unzip PDC\_GUI.zip file and move all content into created directory
* Make necessary changes in Web.config file (details explained bellow in this chapter)
* Using Internet Information Services (IIS) Manager creates PDC instance GUI as a new application of website or create new website.

### Check application tools settings before creation of website/application

1. Application pools -> Target application pool > Advanced settings



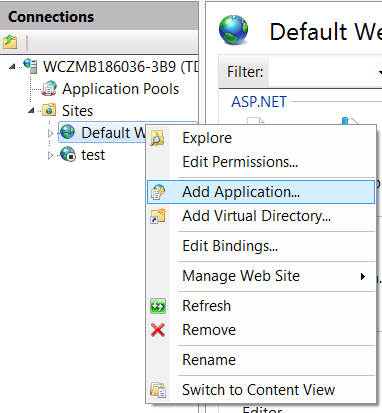
1. Check if .NET framework version is set to 4.0.



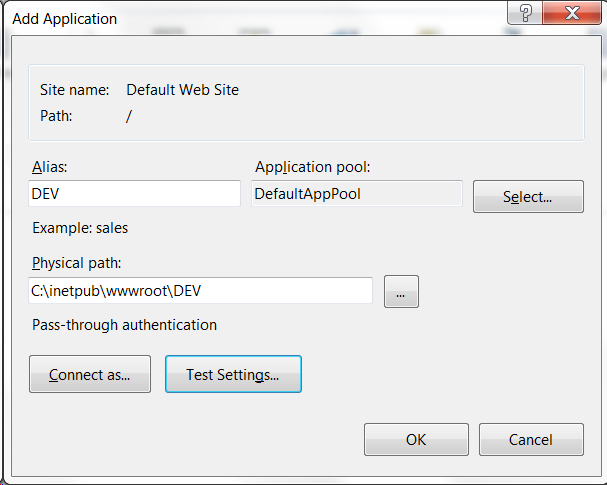
1. Regarding to which 32/64bit ORA client tools are used, select True/False Enable 32-bit applications option.

### Add PDC instance as an application of the website

1. Sites> Default Web Site > Add application



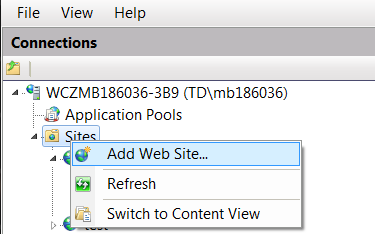
1. Specify URL alias, PDC instance name, select application pool and directory with PDC instance



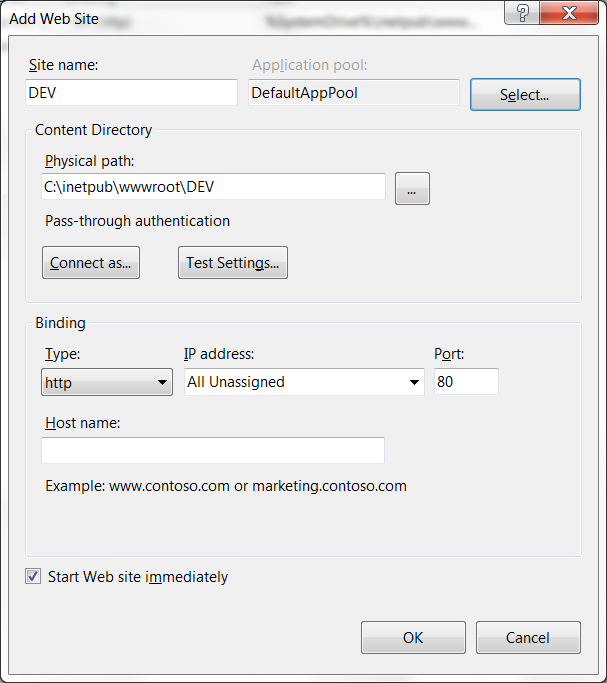
1. Click on OK button and restart target website. Click on OK button and restart target website.

### Add PDC instance as new website

1. Sites> Add Web Site



1. Specify Website name, select application pool, directory with PDC instance and **port on which website will be accessible.**



1. Click on OK button.

### Configuration file:

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<connectionStrings>

...

<add name="PDCOracleConnection" connectionString="Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=hxtstdb8pkg.ux.to2cz.cz)(PORT=1542))(CONNECT\_DATA=(SERVICE\_NAME=UCMT)));User ID=PDC;Unicode=True;Password=xxx" providerName="System.Data.OracleClient" />

</connectionStrings>

...

</configuration>

Use of connection - declarative:

<asp:SqlDataSource ID="SqlDataSource\_gv\_Detail\_lvl1" runat="server"

...

ConnectionString="<%$ ConnectionStrings:PDCOracleConnection.connectionString %>"

ProviderName="<%$ ConnectionStrings:PDCOracleConnection.ProviderName %>" ... >

<SelectParameters>

...

</SelectParameters>

</asp:SqlDataSource>

Use of connection – in code:

using (OracleConnection objConn = new OracleConnection(ConfigurationManager.ConnectionStrings["PDCOracleConnection"].ConnectionString))

{

OracleCommand objCmd = new OracleCommand();

...

try

{

objConn.Open();

objCmd.ExecuteNonQuery();

}

catch (Exception ex)

{

...

}

}

### Security settings on D:\Inetpub\PDCxxx\Web.config file

Use settings:

Properties -> Security -> Full Control ... only for defined „secure“ users and WEB admin (IIS\_WPG). Other „unsecure“ users without any permission !

### GUI Access permissions management

User security model is based on application roles. Simply, user is given role and then, the role is given the particular rights.

The mapping user to role is four-tier to allow the flexibility for future implementation of different authenticating methods (LDAP, …).

The User is mapped to User group (can represent the LDAP group), the User Group is mapped to the Application role and the role is given the effective rights. The overall simplified mapping is shown on the figure bellow.

**GUI\_AUTH**

**GUI\_ACCESS\_USER\_GROUP\_REF**

**GUI\_ACCESS\_GROUP\_ROLE\_REF**

**GUI\_ACCESS\_ROLE\_RIGHT\_REF**

User1

Group 1

Role 1

Right A

Right B

…

Figure 1 PDC User to Role mapping

The PDC Group to Role mapping is implemented as 1:1 relation, but in general the relation can be N:1.

### User Authentication

User is authenticated via login and passport against the database. The credentials are stored in database table GUI\_AUTH. The password is hashed via standard .NET method.

using System.Security.Cryptography;

...

MD5CryptoServiceProvider md5Hasher = new MD5CryptoServiceProvider();

Byte[] hashedBytes;

UTF8Encoding encoder = new UTF8Encoding();

//hash the password

hashedBytes = md5Hasher.ComputeHash(encoder.GetBytes(password));

### Page Authorization

All ASPX pages are under authorization process by default, i.e. all pages are accessible only after login, except the login page and error page. These settings are stored in the Web.Config file of the ASPX NET project. The mentioned part of the configuration file is bellow.

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

...

<system.web>

...

</customErrors>

<authentication mode="Forms">

<forms loginUrl="~/Account/Login.aspx" timeout="2880" />

</authentication>

<authorization>

<deny users="?"/>

</authorization>

</system.web>

<location path="html">

<system.web>

<authorization>

<allow users="\*"/>

</authorization>

</system.web>

</location>

<location path="errors">

<system.web>

<authorization>

<allow users="\*"/>

</authorization>

</system.web>

</location>

...

</configuration>

### User Management Support

Page: AdminUserMngmt.aspx

Procedures:

* PCKG\_GUI.SP\_GUI\_UPDT\_USER\_MNGMT
* PCKG\_GUI.SP\_GUI\_UPDT\_USER\_MNGMT\_DEL
* PCKG\_GUI.SP\_GUI\_VIEW\_USER\_MNGMT

The PDC GUI supports the user management by allowing the administrator to add, modify or remove users from system. Adding user is simple as add the user login and the User group. The user is created with the password same as login (user can change his password afterwards) – procedure: SP\_GUI\_UPDT\_USER\_MNGMT.

Administrator can change the existing User, he can change the User Group – modify the relation. This use the same procedure as user creation but the parameter defining the user password is left empty (NULL).

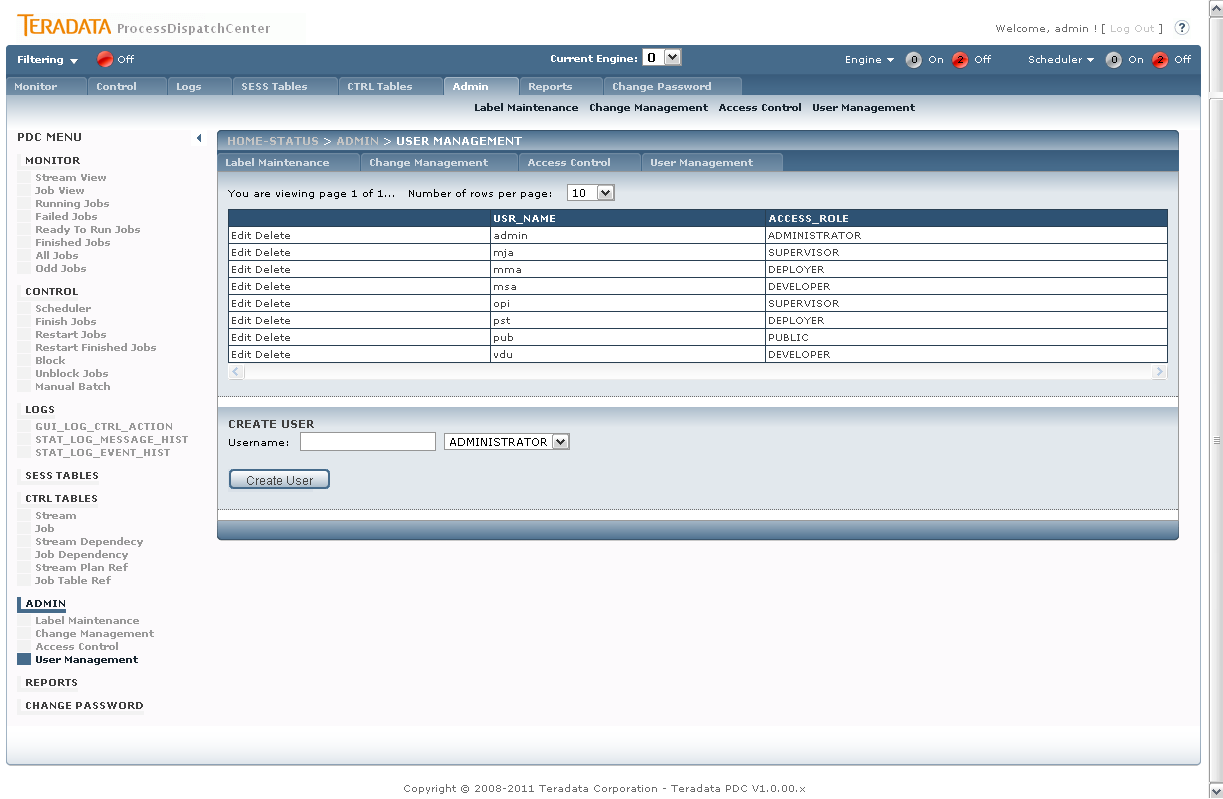


Figure 2 PDC User Management

### Access Control

Page: AdminAccessControl.aspx

Procedures:

* PCKG\_GUI.SP\_GUI\_VIEW\_ACCESS\_ROLE
* PCKG\_GUI.SP\_GUI\_UPDT\_ACCESS\_ROLE
* PCKG\_GUI.SP\_GUI\_UPDT\_ACCESS\_ROLE\_DEL

The access control – user rights – is related to the application role. In order to implement the PDC security model complex, but flexible – the rights are related to the role and the GUI page the right are applied to. The right values are:

* Show – user can view/ has access to the page
* Control – user has access to the control section of the page (e.g. the action buttons on the Running Jobs detail page)

The administrator can change or modify the right assigned to the application role to control the access to the PDC GUI and functionality based on role.

### Password Change

Page: AdminUserPasswdChng.aspx

Procedures:

* PCKG\_GUI.SP\_GUI\_UPDT\_USER\_MNGMT\_PASS

This page is visible to all users/roles by default. The page simply provides the user of password changing functionality based on the current logged user.

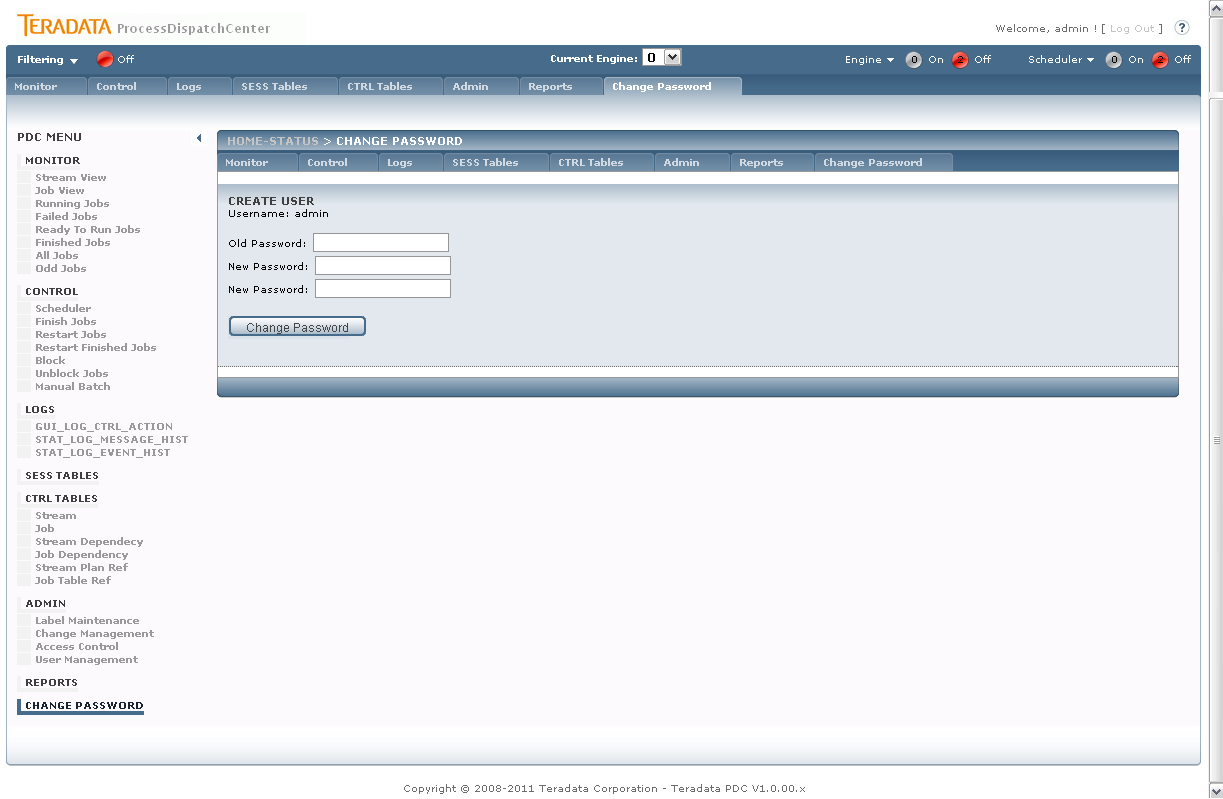


Figure 3 PDC User Management – Password Change

## Task scheduling

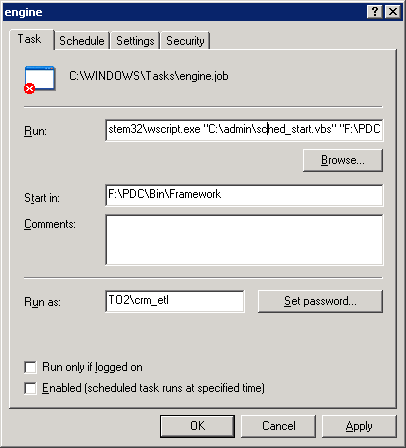
Engine, Initialization and Framework\_checker has to be periodically scheduled.

Use Windows scheduler for scheduling of Engine.bat (every 5 minutes), Framework.bat (every 5 minutes) and Initialize.bat (every day). All three files are located in %PMRootDir%\Bin\Framework directory. Initialization has to be launch for every Engine, in Double Engine environment only once (from one server) for each Engine, preferably on server when Initialization is scheduled. Framework checker is taking care of all Engines, so it’d be scheduled only once per environment.

Use system scheduler tools for setup periodic run of Engine.bat (every 5 minutes), Framework\_checker.bat (every 5 minutes) and Initialization.bat (once per day or periodically for inter-day processing). When “Double Engine” feature is used, schedule Initialization run only on one system or on all system (this solves problem when system is down), but in this case use at least 1 minute schedule shifts to protect starting initialization process on all servers in the same time (this is not valid for inter-day processing).

### Engine

Task:



Run: %WINDIR%\system32\wscript.exe "C:\admin\sched\_start.vbs" "%PMRootDir%\Bin\Framework\engine.bat"

Start in %PMRootDir%\Bin\Framework

Run as: TO2/crm\_etl (with proper password)

Program wscript.exe is coming from standard Windows installation.

VB script sched\_start.vbs can be located anywhere; typically we are placing it in C:\Admin folder.

Script content:

*'--------------------8<----------------------*

*sTitle = "Batch launcher"*

*Set oArgs = WScript.Arguments*

*Set oFSO = CreateObject("Scripting.FileSystemObject")*

*Set oShell = CreateObject("WScript.Shell")*

*If oArgs.Count <> 1 Then*

*' Will die after 10 seconds if no one is pressing the OK button*

*oShell.Popup "Error: You need to supply a file path " \_*

*& "as input parameter!", 10, sTitle, vbCritical + vbSystemModal*

*Wscript.Quit 1*

*End If*

*sFilePath = oArgs(0)*

*If Not oFSO.FileExists(sFilePath) Then*

*' Will die after 10 seconds if no one is pressing the OK button*

*oShell.Popup "Error: Batch file not found", \_*

*10, sTitle, vbCritical + vbSystemModal*

*Wscript.Quit 1*

*End If*

*' add quotes around the path in case of spaces*

*iRC = oShell.Run("""" & sFilePath & """", 0, True)*

*' Return with the same errorlevel as the batch file had*

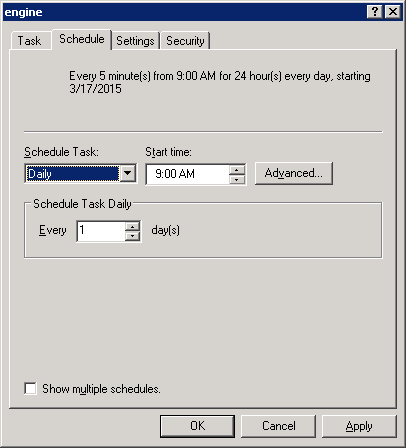
*Wscript.Quit iRC*

*'--------------------8<----------------------*

Engine.bat script contains environment settings and calling of PERL script engine.pl. If several Engines are initiated on one physical server, variable %PMRootDir% cannot be set on system level, in this case %PMRootDir% is set in BAT script. In this case locate Framework folder and use physical path instead of %PMRootDir%. Typically %PMRootDir% is placed in F:\PDC folder.

*Please note that mentioned TO2/crm\_etl user is already existing technology user for aCRM task processing. The reason for reusing already existing user account is the necessity to run Engine by technology user with permissions to execute executive tasks therefore these executive tasks are running as Engine child processes. On other environments, where different executive tasks are launched, we need to use different technology user with necessary permissions. Do not forget that used technology account has to have full permission on Security subfolders for touching passwords and other configuration information, contrariwise other users mustn’t have this right.*

Schedule:

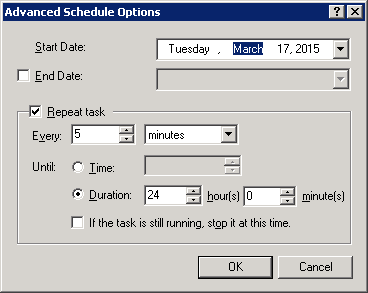


Schedule Task: Daily

Start time: current time

Schedule Task Daily: Every 1 day(s)

Advanced:

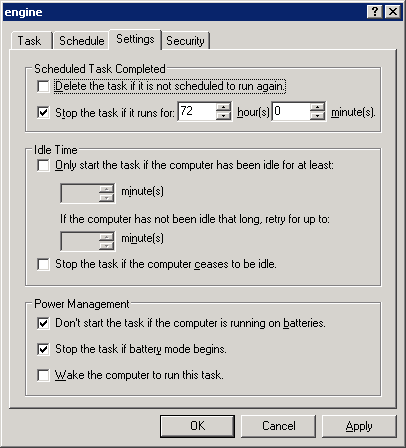


Repeat task: Every 5 minutes

Duration: 24 hours

Settings:

* No change

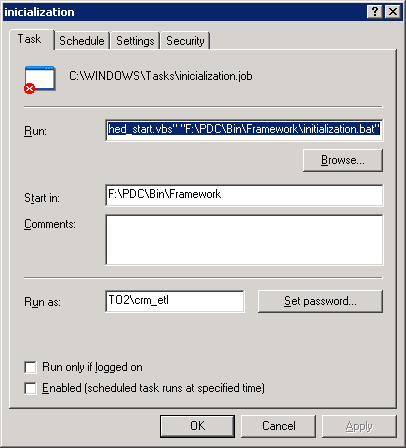


Security:

* No change

### Initialization

Task:



Run: %WINDIR%\system32\wscript.exe "C:\admin\sched\_start.vbs" "%PMRootDir%\Bin\Framework\ initialization.bat"

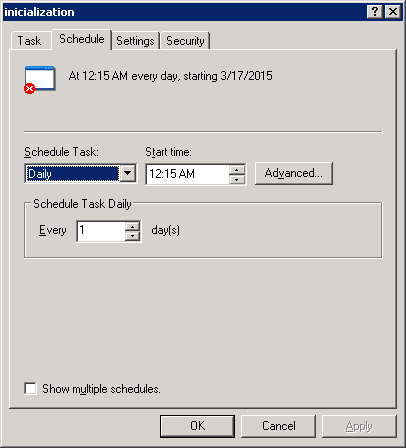
Start in %PMRootDir%\Bin\Framework

Run as: TO2\crm\_etl (with proper password)

Initialization.bat script contains environment settings. If variable %PMRootDir% cannot be set on system level, in this case %PMRootDir% is set in BAT script. In this case locate Framework folder and use physical path instead of %PMRootDir%. Typically %PMRootDir% is placed in F:\PDC folder.

*Please note that initialization for every Scheduler (unique engine\_id) is running only from one server even in “Double Engine” environment. If several identical ETL servers are used in environment (currently not used in O2), when frequently not all servers must run, possibility of scheduling Initialization process on all servers exists. In this case define 5 minutes time shift in startup time. Using the same startup time on all servers will cause a problem.*

Schedule:

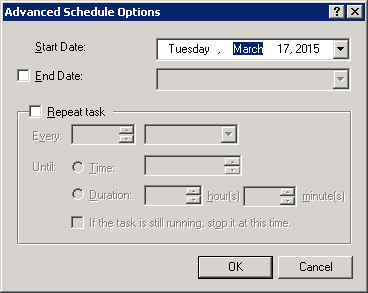


Schedule Task: Daily

Start time: setup as requested

Schedule Task Daily: Every 1 day(s)

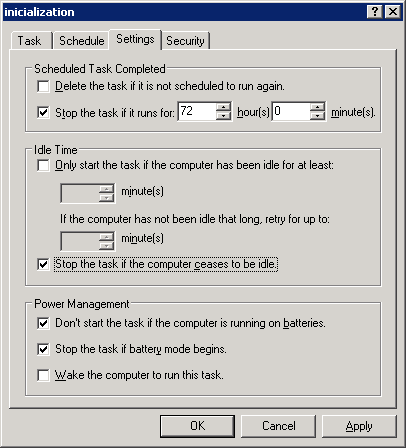
Advanced:



* No change

Settings:

* No change

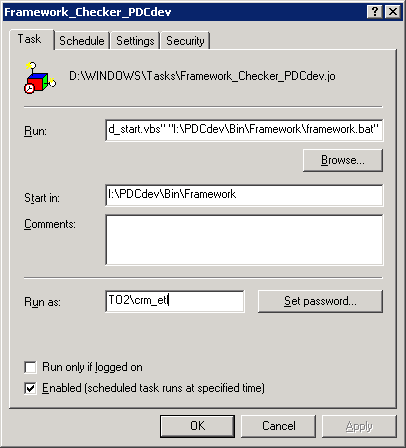


Security:

* No change

### Framework checker

Task:



Run: %WINDIR%\system32\wscript.exe "D:\admin\sched\_start.vbs" "%PMRootDir%\Bin\Framework\framework.bat"

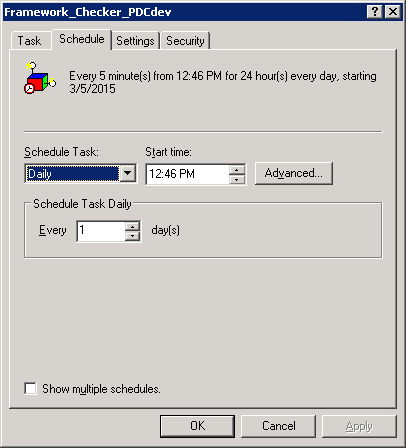
Start in: %PMRootDir%\Bin\Framework

Run as: TO2\crm\_etl (with propper password)

Framework.bat script contains environment settings. If variable %PMRootDir% cannot be set on system level, in this case %PMRootDir% is set in BAT script. In this case locate Framework folder and use physical path instead of %PMRootDir%.

*Please note that only one Framework checker is necessary for each environment (production, test, …). Framework checker checks all Engines in one environment.*

Schedule:

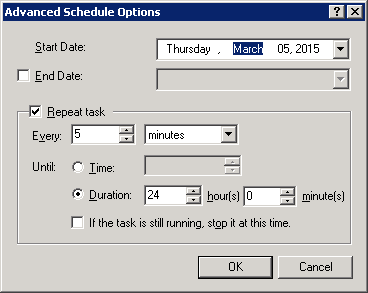


Schedule Task: Daily

Start time: current time

Schedule Task Daily: Every 1 day(s)

Advanced:

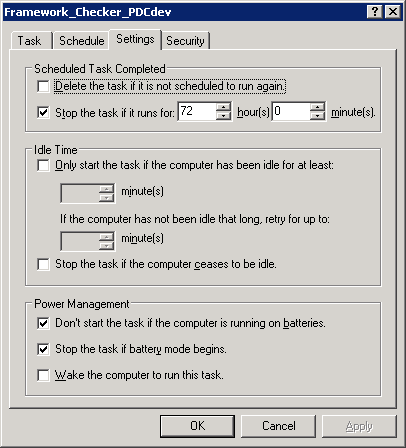


Repeat task: Every 5 minutes

Duration: 24 hours

Settings:

* No change



Security:

* No change

# Environment preparation

## ActiveState PERL Installation (licensed)

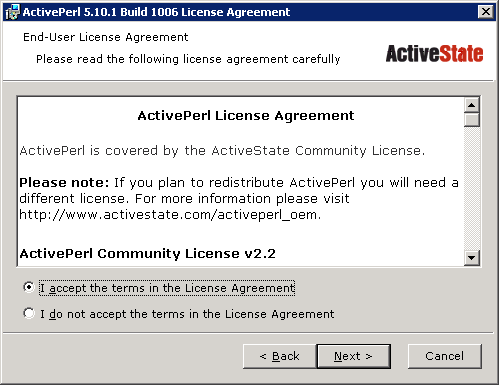
Download the most actual **ActiveState PERL** for used operation system from[**www.activestate.com**](http://www.activestate.com)

*(Please note you’ll need valid license for commercial use).*

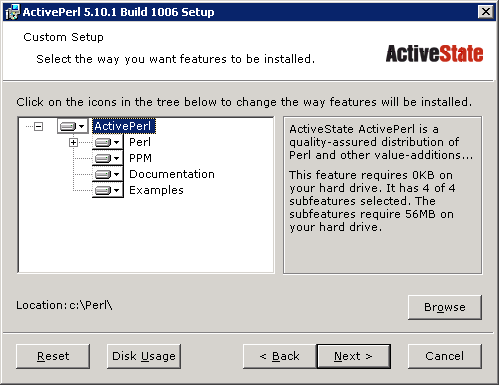
Run installer and go thru installation steps.

ActiveState version 5.14 or above contains all necessary library packages, no additional packages are necessary to download and install, it is not true for older versions of ActiveState PERL. Go thru installation process using default values.

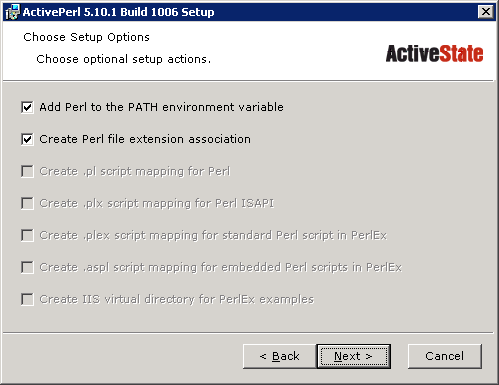




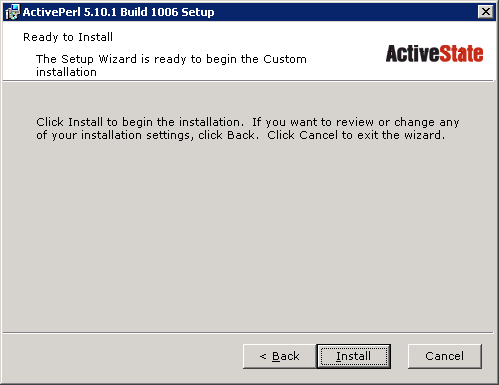
Agree license agreement.



Installation of all components are chosen in default, accepts it.



Lets necessary environment settings are done.



Make installation of ActiveState PERL.



When installation process finish, test installation correctness by typing “perl – v” in command line window.

## Oracle Database Installation and Setup (licensed)

**Oracle database 11g** or above is necessary for Oracle objects creation and metadata storage. Create Oracle database for each environment which will be controlled by PDC framework, e.g. when using production, test and development environment you will need three Oracle databases. Database size depends of metadata history length you will want to hold in database. Standard size is about 1 – 100 GB. There is no special needs regarding tablespace, all objects can be stored in one tablespace. Use default language page for your company and ANSI date/timestamp standard format YYYY-MM-DDBHH24:MI:SS, PDC doesn’t have any other special requirements.

**Example for database creation:**

create tablespace ts\_prod\_pdc

logging

datafile 'C:\oraclexe\app\oracle\oradata\XE\ts\_prod\_pdc.dbf'

size 256m

autoextend on

next 256m maxsize 10240m

extent management local;

create user pdc

identified by pdc

default tablespace ts\_prod\_pdc;

grant dba to pdc;

Level of pdc user rights can be decreased when database import is finished.

Special rights are necessary for procedure failure debugging.

**Special rights settings:**

grant execute on sys.dbms\_pipe to pdc;

grant execute on sys.dbms\_system to pdc;

grant execute on sys.dbms\_utility to pdc;

grant execute on sys.dbms\_output to pdc;

## Oracle Client SW Installation

**Oracle client SW** is necessary for PERL script to metadata touch. We suggest making installation of whole SQL developer package. Download **Oracle Database 11g Release 2 Client for Microsoft Windows** from [www.oracle.com](http://www.oracle.com) and install whole package including command line utilities like “sqlplus”. Use company standards for making this installation. *Please note that respecting of same version of application (PERL, Microsoft IIS, Oracle client, Java SDK and .Net), meaning 32bit vs. 64bit, protects you from future configuration nightmares.*

Download Oracle Client Win64bit installation

During installation choose Administrator option

Finally restart server

Run installer.

## .Net Framework Installation

**Microsoft .Net framework 4.0** or above is necessary for GUI application using Microsoft IIS. Download Microsoft .Net Framework on [www.microsoft.com](http://www.microsoft.com).

Register .Net by command: **aspnet\_regiis –i**

Enable ASP.Net service - put it to automatic

## Microsoft IIS Installation (licensed)

**Microsoft Internet Information Server (Microsoft IIS)** installation is part of Microsoft Server operation system and has to be selected during installation process or can be added afterwards.

Version of OS and ODBC should be same – 32bit or 64bit both.

1. Assuming it has been installed, make sure .Net 4 Framework is registered for ASP.Net by executing:

aspnet\_regiis -i

1. Find out what IIS extensions are enabled (0) and disabled (0) by executing teh following command:

C:\Windows\System32 cscript iisext.vbs /ListFile

1. If the .Net 4 Framework is disabled you will need to run the following command:

cscript iisext.vbs /EnFile C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319\aspnet\_isapi.dll

and verify that you now see a ’1′ in the status of the desired Framework.

1. Now navigate to your Framework version folder and run the following command:

aspnet\_regiis -lv

to list the status and installation path of all versions of ASP.NET that are installed on the computer. Notice the (Root) note that specifies this is the default Framework.

1. Execute the following command:

aspnet\_regiis -lk

to list the path and version of all IIS metabase keys where ASP.NET is mapped.

1. Note in the attached image that the desired application DSMapApp is still targeting the 2.0 framework EVEN THOUGH running aspnet\_regiis -i is supposed to switch all apps to the specified Framework.
2. To switch a particular application’s targeted Framework we’ll need the -s switch to Update scriptmaps and application-pool assignments for the specified application and for all sub-applications.:

aspnet\_regiis -s W3SVC/1/Root/[AppName]