

User Manual



An der Autobahn 18
D-33311 Gütersloh

www.arvato-systems.de | www.e-eis.com

Inhalt

1	Part I: Introduction	4
1.1	Chapter 1	4
1.1.1	Introducing PDC	4
1.2	Chapter 2	5
1.2.1	Package content	5
1.2.2	Software package page	6
1.3	Chapter 3	7
1.3.1	PDC Structure	7
1.3.2	Installation	8
2	Part II: Using PIT	10
2.1	Chapter 4	11
2.2	Starting PIT	11
2.3	Connecting to a cluster	11
2.4	Repositories	12
2.4.1	Defining a repository	12
2.4.2	Removing a repository	13
2.4.3	Editing a repository	13
2.5	Versions	14
2.5.1	Adding a new version	14
2.5.2	Editing a version	15
2.5.3	Removing a version	16
2.6	Configurations	17
2.6.1	Adding new configurations	17
2.6.2	Editing a configuration	18
2.6.3	Changing the config file contents	18
2.6.4	Viewing config file contents	19
2.6.5	Removing a configuration	19
2.7	Machines	20
2.7.1	20
2.7.2	Adding a machine	20
2.7.3	Editing machine details	21
2.7.4	Removing a machine	21
2.8	Processes	22
2.8.1	Adding processes	22
2.8.2	Editing process setup	23

2.8.3	Removing processes.....	24
2.8.4	Start / Stop / Restart.....	24
2.9	PDC Configuration Options.....	25

1 Part I: Introduction

1.1 Chapter 1

1.1.1 Introducing PDC

What is “PDC” ?

PDC stands for ProcessDeployControl, and as such is a distributed software management system intended to be used for deployment and control of server processes on clusters of server machines. PDC consists of an administration program called “PIT” and a controller software named “LocalController” which runs on every system in the server cluster.

A cluster is comprised of one or several machines with a running LocalController on each, one central PostgreSQL database for configuration and dynamic data storage and at least one Subversion repository where the server software is stored.

The administration program PIT connects directly to the database. It has functionality for the management of machines, software versions, configuration sets and the processes themselves.

The LocalControllers also connect to the database and act upon the configuration made by PIT, deploying, starting, stopping, watching processes, replacing one software version by another and reporting back to the database for status reports in PIT.

The processes under control by LocalController make use of the PDCDLL, a dynamic link library utilized by the processes themselves that connects to the PDC database, fetches commands from it and reports back. The LocalController will make use of the data written by the PDCDLL to determine the state of each process and act upon it and the PDCDLL will report commands back to the process so it can safely shutdown on request.

1.2 Chapter 2

1.2.1 Package content

1.2.1.1 Actual platform and requirements for PDC

Microsoft Windows ® XP or XP Pro

Dot net framework 2.0

Pentium/Athlon 2.0 Ghz processor

1 GB RAM

100MB free Disk Space

Internet connection 256 Kbit or higher

1.2.2 Software package page

This page is intended left blank to hold the software CD for the physical distribution.

tinCat PDC

The Network System.

Please look for the version, language and notes on the CD
for the latest release.

1.3 Chapter 3

1.3.1 PDC Structure

1.3.1.1 Machine and LocalController

The LocalController is PDC's main work horse. It runs on all machines that shall be under PDC's control. One LocalController appears as one machine in the system. Although it is possible to run several LocalControllers on one physical machine and make it appear as two or more machines in the system, this is not recommended.

1.3.1.2 Version

A version is a version of your software meant to be deployed by PDC. There can be as many or as few versions as needed. A version is nothing else than a centrally stored directory tree in addition to a path to the executable within this directory. Two different versions can hold entirely different programs and their data and don't need to be related in any way.

1.3.1.3 Process

Process is the term for a program under control by PDC. Each process has a "Version" it operates on. When a process is created, one of the previously defined "Versions" has to be selected for this process. This version is then copied to the machine designated to run the process. A process can be created, removed, started, stopped, it's underlying version can be replaced etc.

1.3.1.4 Configuration

Each process also has a configuration. A configuration contains the commandline parameters for a process, as well as the contents of a text configuration file and a filename that specifies where this file is to be stored when a process is deployed. (e.g. "setup.cfg")

1.3.1.5 PDC Dll

The PDC DLL is a dynamic link library that is linked into and used by the processes under control by PDC. It is used to report statistical data of the process to the database and fetch and route commands from the database to the process. All programs under PDC control are supposed to make use of this DLL.

1.3.1.6 Cluster

A cluster in PDC terms is a collection of server machines that use the same database for operation. More exact, a cluster is a collection of LocalControllers that connect to the same database. It is possible to have two LocalControllers on one machine with each of them connecting to different databases, or two LocalControllers connecting to the same database and as such having one physical machine appear as two in the PDC system. It is recommended though to only have one LocalController per machine. In addition, a cluster needs at least one subversion repository. These repositories are used to store the software versions.

1.3.1.7 PIT

PIT is PDC's administration GUI.

1.3.2 Installation

1.3.2.1 Setting up the repository

A SVN (Subversion) repository is used to hold the software under control by PDC. The repository must be accessible by all server machines in the cluster and by the machines running the administration program “PIT”. It can be username/password protected. Repositories can be hosted via any means described in the official subversion documentation.

1.3.2.2 Setting up a new machine in the cluster

To add a new machine to the server cluster the LocalController and all it's files need to be copied to the machine in question. A dbaccess.cfg file needs to be copied from the PIT program's directory, as it contains the username and encrypted password to connect to the database. Delete all the database entries except the one for the cluster of the new machine, and change the 'localmachineid' entry so that it contains the ID of the machine this LocalController represents, which can be seen in PIT under the 'Machines' tab. Start the LocalController.exe and see that it does not complain about a failed database connection or an invalid machine ID.

1.3.2.3 Installing PIT on your computer

The AdminGUI PIT is a C# (c-sharp) application which needs the Microsoft .NET (dot net) framework of the right version to run properly. Please check your computer for the version of the .NET framework (See requirements or readme file) or if you are not sure, install the included Microsoft .NET framework redistributable.

1.3.2.4 Security advice

PDC's user management is wholly done by the PostgreSQL database. So for a PIT user to log into the system and perform changes, he needs to supply a database username and password. The processes he creates will use the same credentials as the user who created them.

2 Part II: Using PIT

2.1 Chapter 4

2.2 Starting PIT

To start PIT simply launch the “PIT.EXE”.



2.3 Connecting to a cluster

On the left hand side you will find the list of available clusters. If this is the first time you launch PIT, there will be no cluster connections available. Press the “Add” button to create a new cluster connection and fill in the data on the right hand side of the window.

2.3.1.1 Serveraddress

This is the address of the PostgreSQL database server.

2.3.1.2 Databasename

The name of the database inside the database server.

2.3.1.3 Username / Password

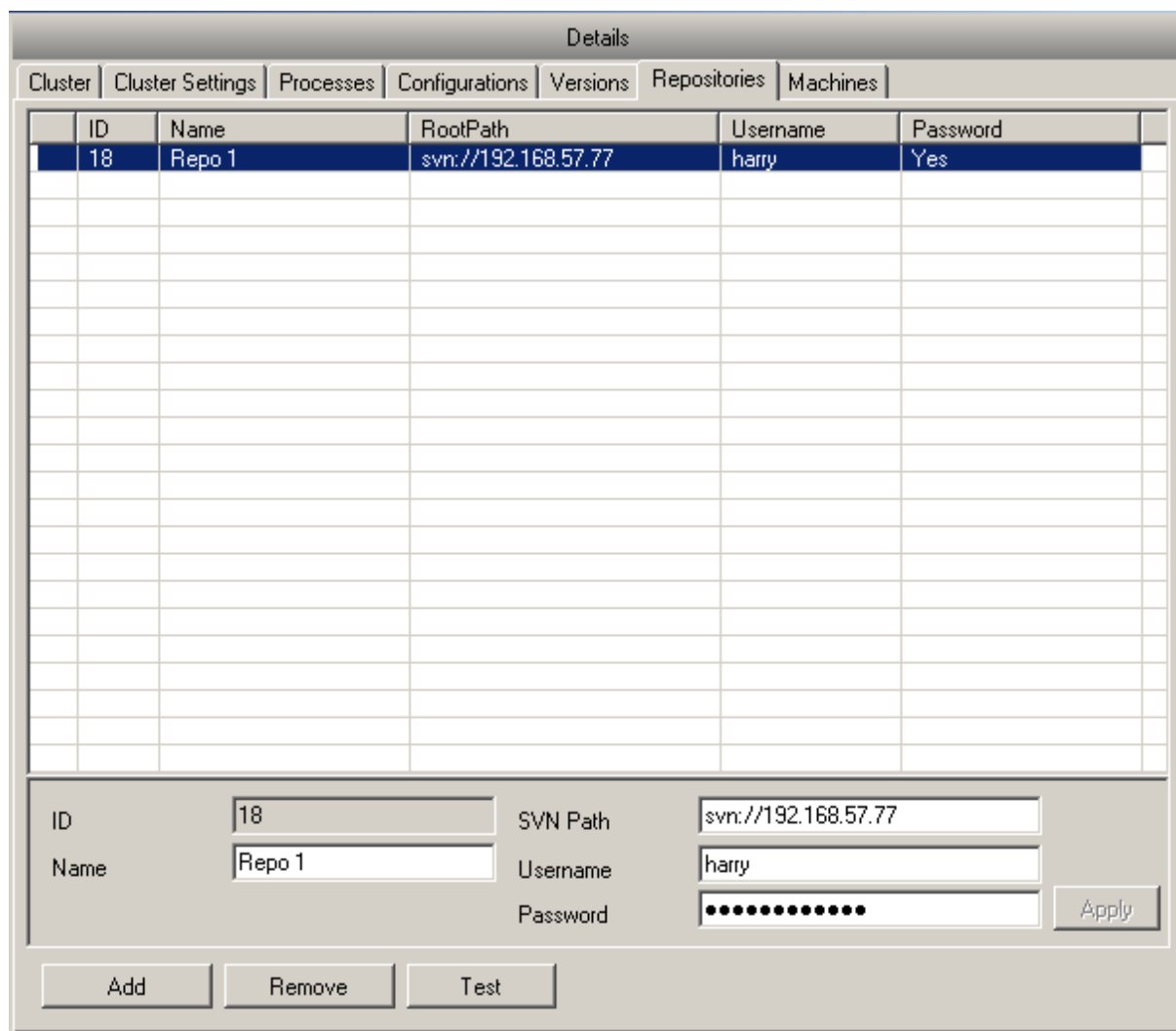
The username and password for the database.

After everything has been filled in, you can press the “Connect” button to connect to the selected database. After a short delay, a green icon should appear next to the database entry in the list. Please note that you can connect to as many clusters simultaneously as you like, but that all operations and all outputs within PIT only relate to the database/cluster currently selected in the list on the left hand side.

To remove a cluster connection again, simply select it and press the “Remove” button.

2.4 Repositories

One of the first things to do when starting up a new cluster is to define a repository connection so that the cluster has a source for the to be deployed software.



Details

Cluster Cluster Settings Processes Configurations Versions **Repositories** Machines

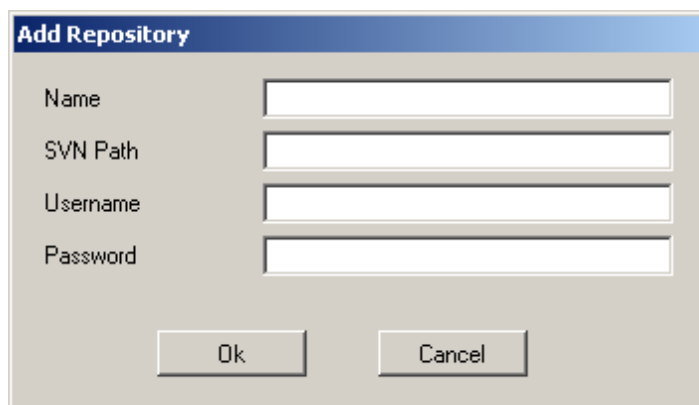
ID	Name	RootPath	Username	Password
18	Repo 1	svn://192.168.57.77	harry	Yes

ID: 18 SVN Path: svn://192.168.57.77
 Name: Repo 1 Username: harry
 Password: ●●●●●●●● Apply

Add Remove Test

2.4.1 Defining a repository

Press the “Add” button to define a new repository connection.



The image shows a dialog box titled "Add Repository". It contains four input fields: "Name", "SVN Path", "Username", and "Password". At the bottom of the dialog, there are two buttons: "Ok" and "Cancel".

2.4.1.1 1) Name

The repository name is just a freeform name to identify the repository in other places in the PIT program. It has to be unique.

2.4.1.2 2) SVN Path

This is the subversion repository path such as “svn://192.168.5.33”.

2.4.1.3 3) Username and Password

If an authentication is required to access the repository, enter username and password in the respective fields.

Press “Ok” to add the repository.

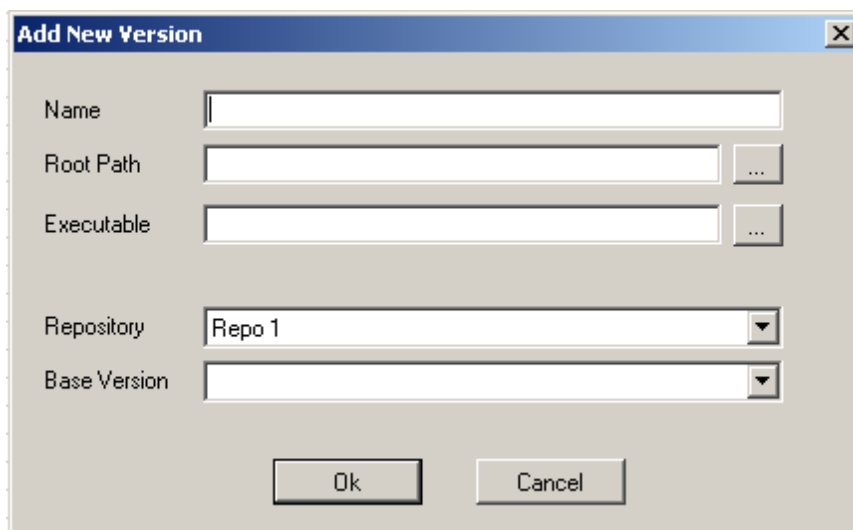
To test if the repository connection works, select the newly added repository in the list and press the “Test” button. After a short delay a result message box appears and reports success or failure. In some failure cases it can take several seconds for the message to appear.

2.4.2 Removing a repository

To remove a repository, select it in the repository list and press the “Remove” button. Repositories can only be removed if there are no “Versions” in the database that are located in this repository.

2.4.3 Editing a repository

If the repository location, the username or password change you can edit these details by clicking on the repository in the list and changing the data at the bottom of the window. When you have finished editing, press the “Apply” button to submit the changes to the database. All following operations on the repository by PDC will use the new data.



2.5.1.1 1) Name

The name of the new version such as “Server 1.0”, “Server Debug 1.1” etc. Must be unique.

2.5.1.2 2) Root Path

Enter the local path (or press the “...” button) of the software on your machine here. This path and all it's files and subdirectories will be uploaded to the repository.

2.5.1.3 3) Executable

Choose an executable within the root path that has to be started to get this version running. This is the exe that will be started when the version is deployed to other machines.

2.5.1.4 4) Repository

Choose the repository where this software version is to be uploaded to.

2.5.1.5 5) Base Version

Here you can choose from all versions hosted in the selected repository. If a base version is selected, the currently added version will be stored as delta/diff to the base version, thus saving storage space in the repository. So if you add a software totally different from all other versions in the repository, no base version should be selected. If on the other hand the new version is similar to the base version (lots of files are the same), then only the changed files will be uploaded and stored in the repository.

Press “Ok” to begin the upload process. For large amounts of data, this will of course take some time.

2.5.2 Editing a version

After a version is created, you can change it's name or the executable path by selecting the version in the repository and editing the fields in the lower part of the screen. When you are done,

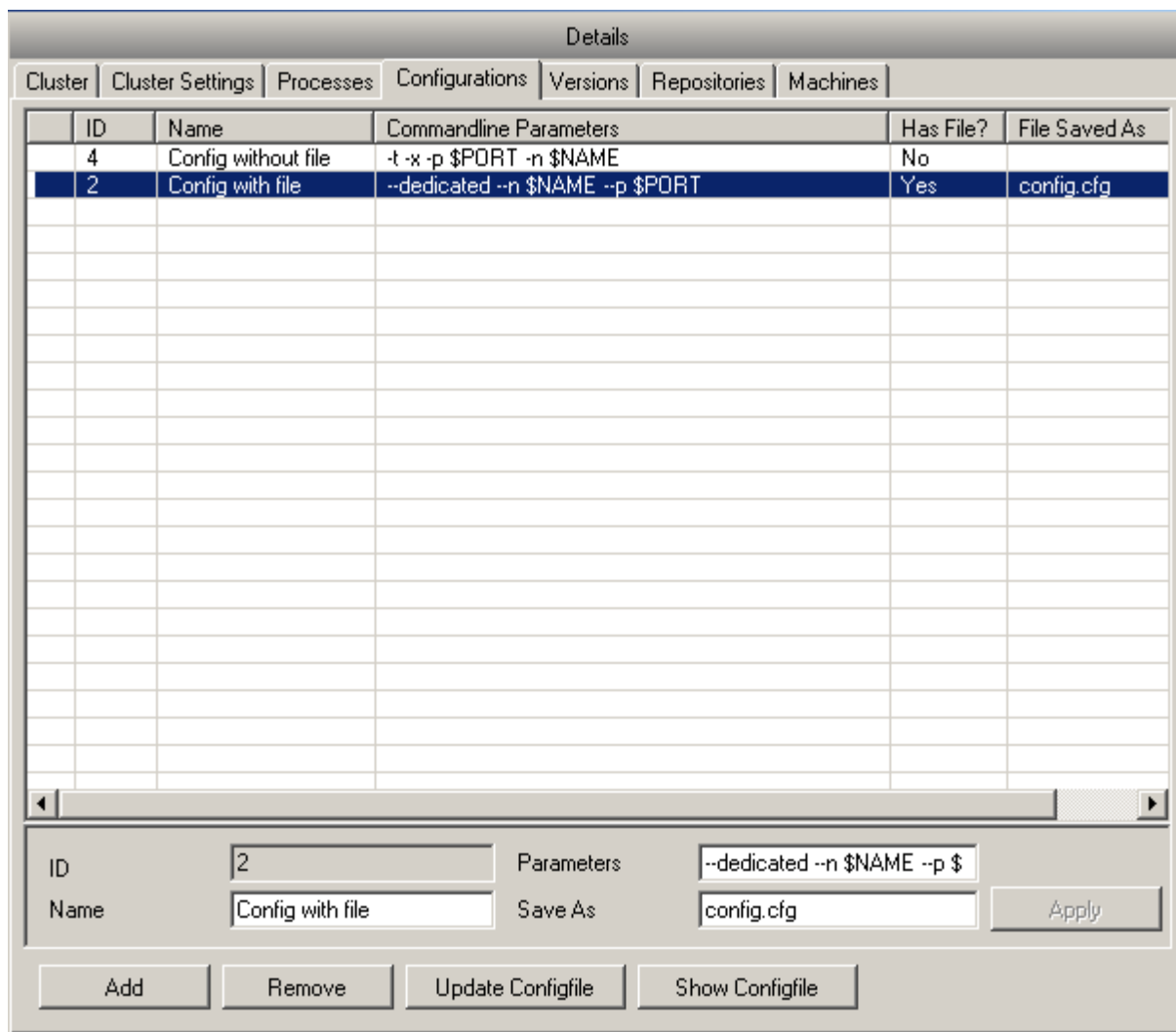
press “Apply”. Note that changing the executable path of versions that are currently in use by deployed processes will shutdown these processes, redeploy and restart them.

2.5.3 Removing a version

To remove a version select it in the version list and press the “Remove” button. Note that you can only remove unused versions. To remove a version in use by processes the respective processes need to be removed first.

2.6 Configurations

Each process needs a configuration to go along with it. A configuration is a commandline and a config file, both of which are passed on to the processes when it is started.



The screenshot shows a window titled "Details" with several tabs: Cluster, Cluster Settings, Processes, Configurations, Versions, Repositories, and Machines. The "Configurations" tab is active, displaying a table with the following data:

ID	Name	Commandline Parameters	Has File?	File Saved As
4	Config without file	-t -x -p \$PORT -n \$NAME	No	
2	Config with file	--dedicated --n \$NAME --p \$PORT	Yes	config.cfg

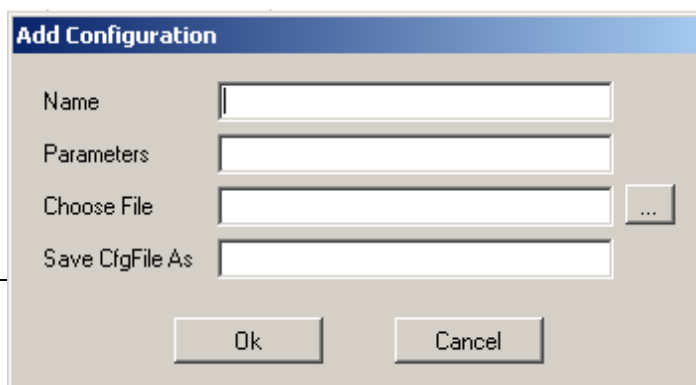
Below the table is a form for editing a configuration. The fields are:

- ID: 2
- Name: Config with file
- Parameters: --dedicated --n \$NAME --p \$
- Save As: config.cfg

Buttons at the bottom include: Add, Remove, Update Configfile, Show Configfile, and an Apply button next to the Save As field.

2.6.1 Adding new configurations

To add a new configuration press the “Add” button.



The "Add Configuration" dialog box contains the following fields:

- Name: [Empty text box]
- Parameters: [Empty text box]
- Choose File: [Empty text box] with a browse button (...)
- Save CfgFile As: [Empty text box]

Buttons at the bottom are: Ok and Cancel.

2.6.1.1

2.6.1.2 1) Name

A unique name for the configuration.

2.6.1.3 2) Parameters

Commandline parameters to be passed to the exe when the process is started. \$PORT and \$NAME in the commandline are substituted by the process' name and configured network port, so that you don't need to create a new configuration for each process just because it should run with a different name or port.

2.6.1.4 3) Choose File

Enter a path to a local text configuration file here or use the “...” button to open a file browser. The contents of this text file are stored in the database along with the configuration.

2.6.1.5 4) Save CfgFile As

Enter the path and name under which the configuration file contents should be saved when the process is deployed. This path is relative to the version's binary path. If the executable is located in “bin/myserver.exe” then the config will be saved as “bin/config.cfg” if you type “config.cfg” into this field. You can also enter things like “../config.cfg” to have the config saved one level above the executable path.

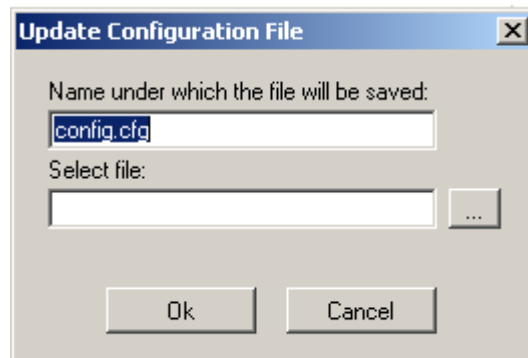
Press “Ok” to submit the new configuration.

2.6.2 Editing a configuration

To edit a configuration you have to select it in the configuration list and then edit the respective fields in the bottom of the window. After editing is done, press the “Apply” button to submit the changes. All changes except to the name of the configuration will restart all processes that use this configuration.

2.6.3 Changing the config file contents

To change the contents of the config file select the configuration in question and press the “Update Configfile” button.



The fields in the appearing dialog have the same meaning as in the “Add Configuration” dialog.

2.6.4 Viewing config file contents

To show the contents of the configfile select the configuration from the configuration list and press the “Show Configfile” button.

2.6.5 Removing a configuration

To remove a configuration select it in the configuration list and press the “Remove” button. Only unused configurations can be removed.

2.7.2.1

2.7.2.2 1) Name

A unique freeform name for the machine.

2.7.2.3 2) Address

Additional info for the server. It has no effect to the PDC system, but is displayed in various places in the GUI. You can enter a machine name, IP address, workgroup name or anything else to identify the location of the machine.

2.7.2.4 3) Deploypath

The path on the machine to which all software is deployed. If this is empty, everything is installed to %TEMPDIR%.

2.7.3 Editing machine details

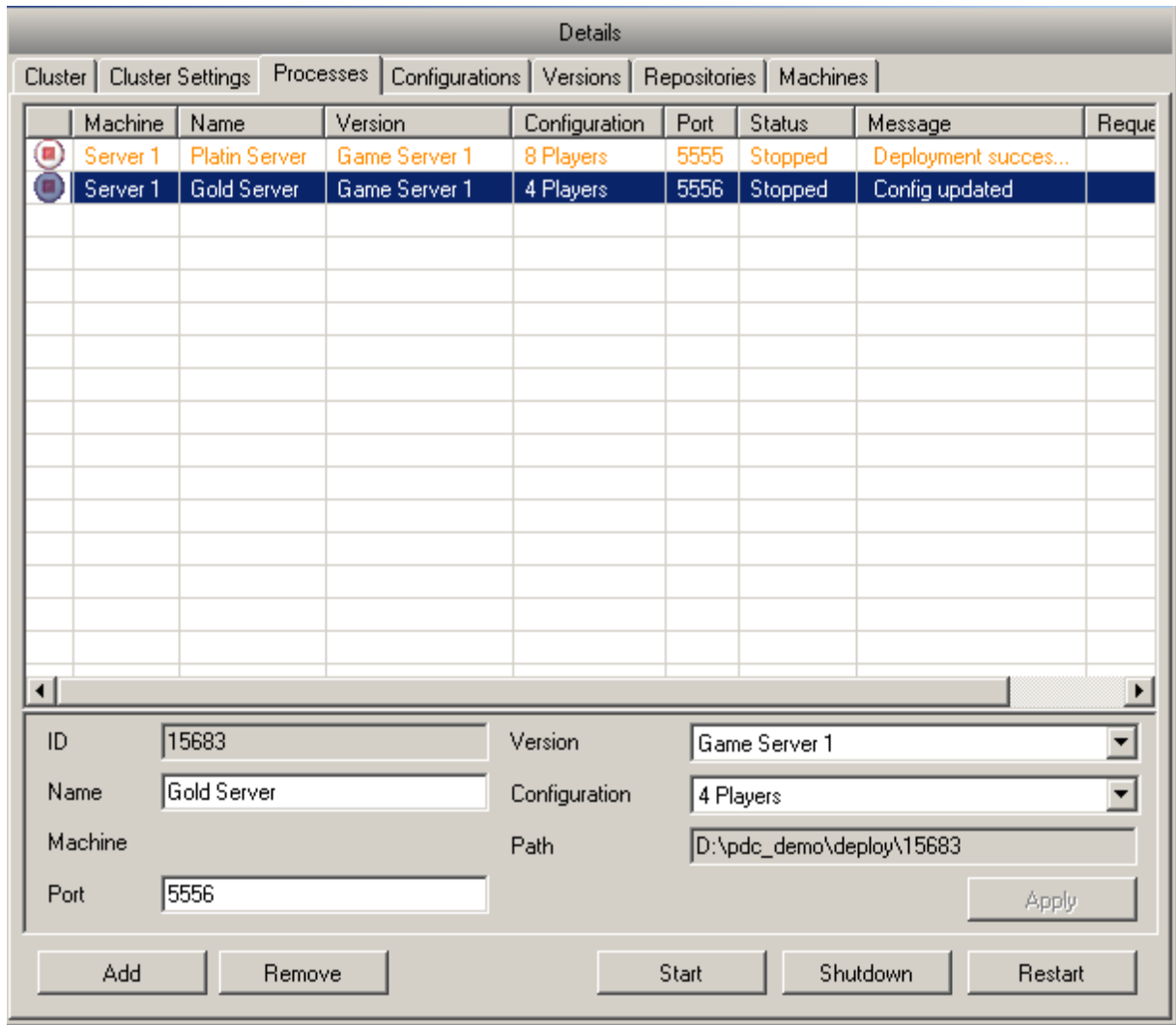
To edit machine details select a machine in the machine list, edit the textfields at the bottom of the screen and press “Apply” when done. Machine name and address can be changed at will, but deploy path can only be changed if there are no processes assigned to the machine.

2.7.4 Removing a machine

To remove a machine select it in the list and press the remove button. If the machine is currently not responding, you will be asked if the remove should be forced. In that case, the machine and all its processes will be deleted from the database without cleanup on the machine itself. This is useful if the machine has already been removed from the network. If on the other hand it is expected to come back online, you can choose to use the normal remove process, in which all processes will be deleted from the machine followed by an automatic shutdown of the LocalController.

2.8 Processes

Each process is one instance of a version that can be started, stopped and more. PDC automatically controls processes to a certain extent. If a process does not update its timestamp for a configurable amount of time, it will be killed and restarted. If a process exits without being requested to, it will be restarted and the crashcounter will be increased. The higher the crashcounter, the longer PDC will wait until it restarts the process. After a default of ten restarts, PDC will stop restarting it.

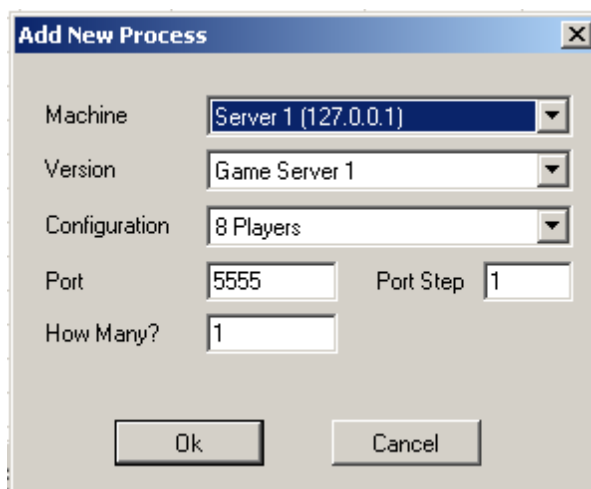


Machine	Name	Version	Configuration	Port	Status	Message	Request
Server 1	Platin Server	Game Server 1	8 Players	5555	Stopped	Deployment succes...	
Server 1	Gold Server	Game Server 1	4 Players	5556	Stopped	Config updated	

ID	<input type="text" value="15683"/>	Version	<input type="text" value="Game Server 1"/>
Name	<input type="text" value="Gold Server"/>	Configuration	<input type="text" value="4 Players"/>
Machine		Path	<input type="text" value="D:\pdc_demo\deploy\15683"/>
Port	<input type="text" value="5556"/>	<input type="button" value="Apply"/>	

2.8.1 Adding processes

To add a process go to the “Processes” tab and press the add button.



2.8.1.1 1) Machine

Here the target machine for the process should be selected. This is the machine on which the process will run once it is deployed.

2.8.1.2 2) Version

Select the version of your software that should be deployed to the machine.

2.8.1.3 3) Configuration

Select the configuration for the process.

2.8.1.4 4) Port

Choose the network port on which the process should listen for incoming connections. This port will be passed through commandline parameter substitution to the process when it is started. See the chapter about configurations for more information. When more than one process is being added, this port is the port for the first machine.

2.8.1.5 5) Port step

When more than one process is being added, this issues the gap between the ports of two generated processes. If Port is set to 10000, Port Step is 2 and you are adding 5 machines, these ports will be assigned: 10000, 10002, 10004, 10006, 10008.

2.8.1.6 6) How many?

Enter the number of processes to generate.

2.8.2 Editing process setup

You can edit the setup of a process or several processes at once by selecting them in the process list. Make your changes in the bottom of the screen and press the Apply button when done. You can move processes to other machines, change the underlying version, change the config, change port and rename them. Note that all of these changes require the processes to at least be res-

tarted, in the case of changing version they will be redeployed and in the case of moving them to another machine, they will be deleted from the old machine and redeployed on the new one.

2.8.3 Removing processes

To remove one ore more processes select them in the process list and press the remove button. They will be stopped and removed from the hosting server.

2.8.4 Start / Stop / Restart

Processes can be manually stopped, started and restarted by selecting them and pressing the respective butons at the bottom of the window.

2.9 PDC Configuration Options

You can configure various PDC parameters through the “Cluster Settings” tab.

Parameter	Value	Unit
Machine response timeout	15	seconds.
Process response timeout	5	seconds.
Restart hanging process after	30	seconds.
Process start failed after	15	seconds.
Restart crashed processes	10	number of times.
Restart crashed processes after	10	* NumberOfRetries ² seconds.
Process shutdown forced after	2	seconds.

2.9.1.1

2.9.1.2 1) Machine response timeout

This determines after how many seconds a machine is considered to be not responding. I.e. If the machine does not report to the database within this amount of time, the GUI will present it as gone and all it's processes get the status “Unknown”.

2.9.1.3 2) Process response timeout

This determines after how many seconds of not reporting to the database a process is considered to be hanging.

2.9.1.4 3) Restart hanging process after

If a process is 'Hanging', it will be killed and restarted automatically after this amount of seconds.

2.9.1.5 4) Process start failed after

This is the time in seconds the process is allowed to use for start up. If a freshly started process does not report to the database after this amount of time, the startup is considered to be a failure. The process enters the 'CantStart' state and will stay there. Manual restart is needed to try another start or change the software.

2.9.1.6 5) Restart crashed processes

This configures how often PDC tries to restart crashed processes. When this limit is reached, the process will be ignored until it is manually (re)started again.

2.9.1.7 6) Restart crashed processes after

This configures a quadratic time factor for the process restart interval with the formula $\text{NumberOfTimesCrashed}^2 * \text{TimeFactor}$. If the time is set to 10 seconds, the first crash will restart the process after $1^2 * 10 = 10$ seconds. The 2nd crash will restart the process after $2^2 * 10 = 40$ seconds and so on.

2.9.1.8 7) Process shutdown forced after

When a process is stopped by PDC for whatever reason, it is first asked to shut itself down. This determines the timespan the process has for fulfilling this task. Once the time is up, PDC will kill the process.

The “Save” button writes the new values to the database and will thus be used by the cluster from this point on. The “Reload “ button can be used to refill the values in the GUI with the values from the database.