www.project-open.com

]project-open[V3.0

Operations & Maintenance Guide

Klaus Hofeditz and Frank Bergmann,
V1.0, 16.8.2005

INDEX

1 About this Document

1.1 Version

Version: 1.0, 16.8.2005

Author: Klaus Hofeditz and Frank Bergmann

Status: Advanced Draft

1.2 Scope

This manual describes how to operate and maintain a **]project-open[** system.
The manual does not describe the initial installation of the system nor the initial configuration of the system. Please see the PO-Installation-Guide and the PO-Configuration-Guide for these purposes.

This guide describes operations and maintenance for Windows systems. However, the same processes apply to Unix/Linux system with minor changes in the command line parameters.

1.3 Audience

This manual is written for system administrators of **]project-open[**. However, most of the describe processes can be executed by any power user.

2 About Operations & Maintenance

Operations & Maintenance processes are necessary to keep a software application running during the time that it is used in a company. The complexity of these processes varies heavily with the size of the company:

SOHO Companies (<3 Users)

Most of this manual is overkill for a home office. The basic principles apply, but the procedures are becoming much easier.

Operations & Maintenance are basically reduced to the periodic use of the “Automatic Software Update Services” (ASUS) that is built into **]project-open[** (see chapter 3). This service is similar to the “Windows Update Service” from Microsoft and doesn’t require many system administration skills.

Small Companies (<10 Users)

Small companies usually have a dedicated System Administrator for the maintenance of PCs and the local area network. This SysAdmin can use the “Automatic Software Update Services” (ASUS) to update the system, similar to SOHO companies.

Larger Companies (<10 Users)

Larger companies will probably have to implement the entire scheme. Senior management should control that the processes are handled correctly, in particular the testing phase on the Staging Server.

3 Notation & Conventions

Text written in Courier with gray shadow, starting with a "#" is program code executed from a "Bash" shell:

```
# echo "This is an ordinary command"
```

This code can be executed in Linux, Solaris, Mac OSX and other Unix-like systems via a normal shell. In Windows please use the "CygWin Bash Shell" command in Start -> Programs -> ProjectOpen or double-click on the C:\ProjectOpen\cygwin\cygwin.bat command.

Text starting with "projop#" indicates database statements:

```
projop# select now(), 'This is a database command';
```

To execute this statements please use the pgAdmin III application and the "SQL" screen (part of the icon bar on the top) or execute "psql projop" on a Bash shell.

Please observe that MS-Word is putting diagonal "double quote" and 'single quotes' in the grey text above. This is wrong and will give you errors. Please use straight double and single quotes when entering commands.

4 Simplified Operations & Maintenance

Operations & Maintenance for **SOHO** and **Small Companies** can be reduced to the use of the ASUS (Automatic Software Update Service). This service is similar to the “Windows Update Service”. Please click on the “Admin” tab or your **]project-open[** installation and then click on “Software Updates” to reach this page.

ASUS is currently (8/2005) free, but we will charge a small monthly change in the future to cover our software maintenance costs.

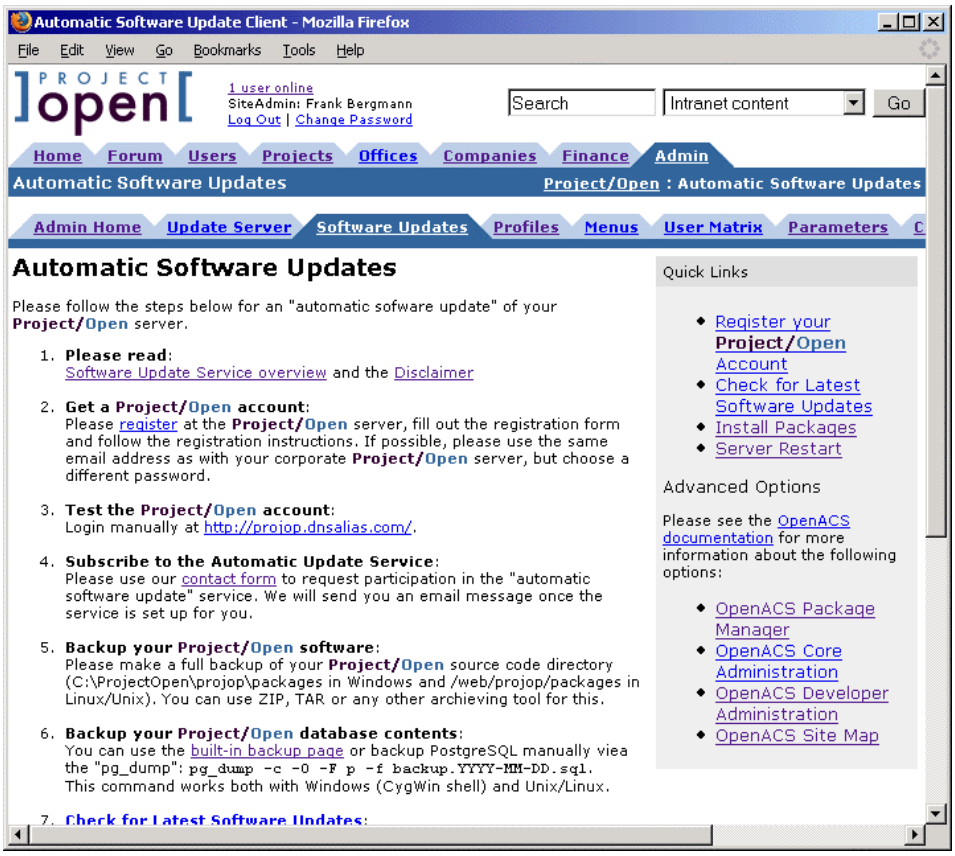


Figure 1: ASUS- The Automatic Software Update Service main screen.

5 Operations & Maintenance Overview

The figure above provides an overview over all processes covered in this manual. The processes will be explained one-by-one in the following chapters.

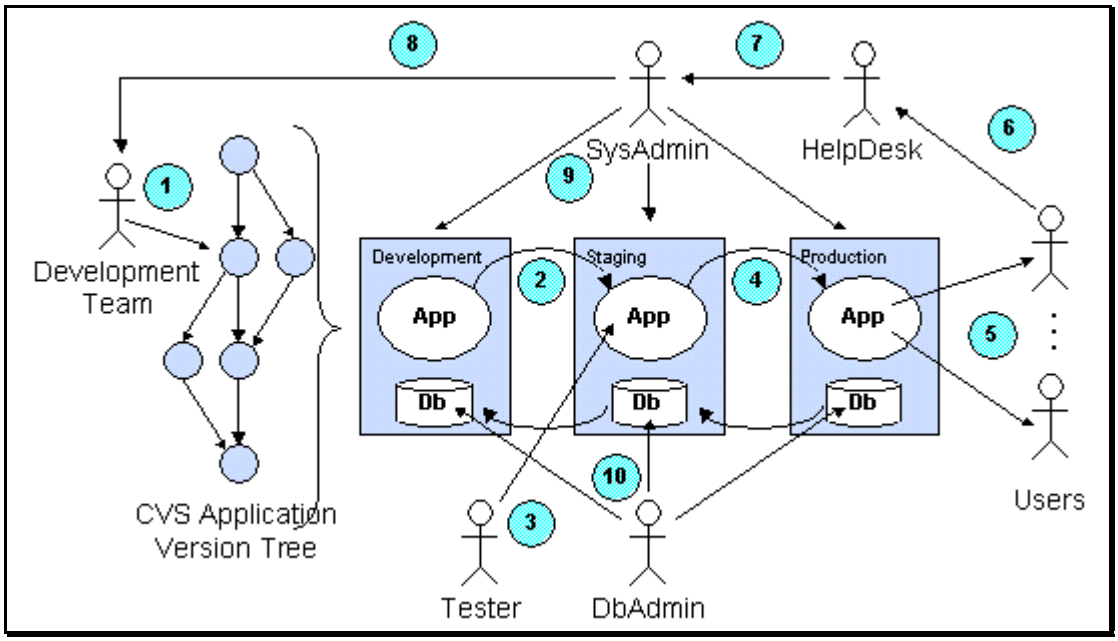


Figure 2: Overview of Operations and Maintenance Processes

The figure is composed of people who are interacting with technical items such as the software application and “server” computers.

5.1 Roles

The figure above uses several “roles” to describe the responsibilities of the people related to a with **]project-open[** system:

- **SysAdmin:**
  Keeps the server running: This should be the most technical person in your company
- **DbAdmin:**
  Keeps the database running: Usually identical with the SysAdmin
- **Tester:**
  Tests system changes: Double-checks the work of the SysAdmin, so it needs to be a different person.
- **HelpDesk:**
  Maintains contact with **]project-open[**: In charge of answering help requests from company’s end users.
- **Development Team:**
  Modifies the application: Performs changes in the application code. This can be performed either in-house or by **]po[**

5.2 Servers

Also the following symbols are used in the figure above to refer to several types of servers:

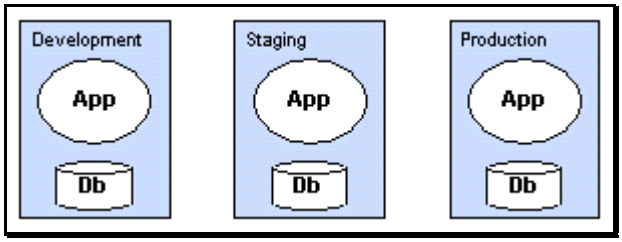


Figure 3: Three different servers to run a single application

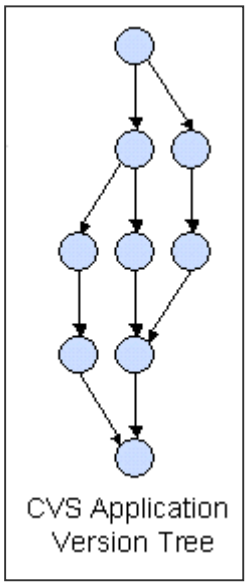
The figure above represents three different servers that are used during the lifecycle of a **]project-open[** application:

- **"Development Server"**:
  The Development Team uses this server in order to fix bugs and to develop new product features. Every software developer usually runs his own development Server. A development server can be any desktop computer running **]po[**. For example you may run a development server in your company if you are experimenting with the system.
- **"Staging Server"**:
  Also called "Testing Server": This server has the only purpose to test the application before it becomes used at the "Production Server". The Staging Server is frequently used as a backup system for the case that the Production Server fails.
- **"Production Server"**:
  Failures of the Production Server may cause financial loss to your company, so your Production Server should be equipped with a RAID disk array and a USB power supply. However, you don't need to buy a new computer for **]po[** because it perfectly OK to run **]po[** together with your file server on the same machine.

5.3 Application Code

The "CVS Application Version Tree" in the figure above represents the **]project-open[** application code.

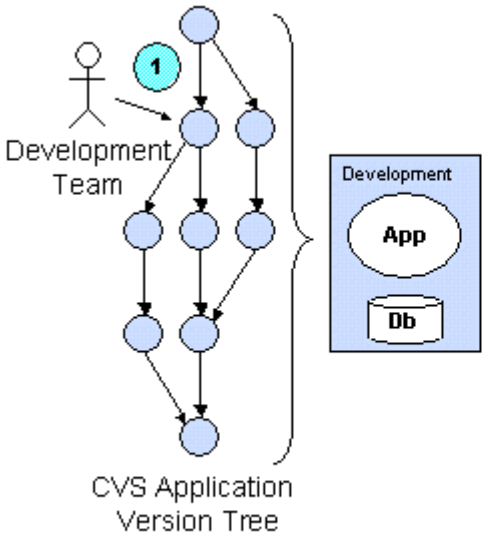
"CVS" is the "Concurrent Versioning System" that allows developers to modify the code just like a Word document with "track changes" enabled. Each circle represents a version of the code with changes from one developer. Circles usually carry version numbers such as V3.0.0.5.6 etc.



6 Bug Fixes and Updates

From time to time you may have to update your system in order to incorporate bug fixes or to take advantage of new product features. Each update is composed of the following stages:

6.1 Software Development



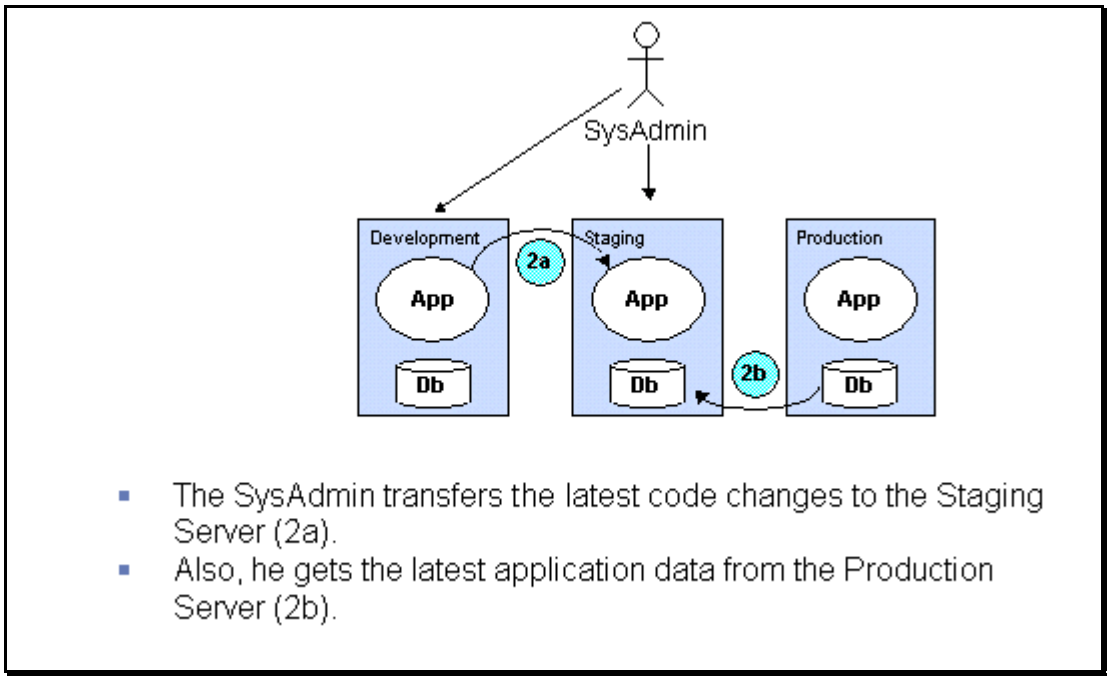
- The Development Team modifies the system in order to implement new features. The new features are tested at the Development Server.
- The Development Team can work on different versions of the application simultaneously (CVS version tree).

Software development for **]po[** is done using a simple text editor if a **]po[** system is running on your computer. Please see the “Learning **]po[**“ page at http://www.project-open.org/ for details.

The **]po[** core team uses “CVS” for software versioning and change management and the www.sourceforge.net/projects/project-open/ online community to coordinate the development. Please let us know if you want to participate in the development.

6.2 "Staging"

The "staging" process has the purpose to create a testing environment on the Staging Server that is as close as possible to the Production Server. Staging consists of two steps:



- The SysAdmin transfers the latest code changes to the Staging Server (2a).
- Also, he gets the latest application data from the Production Server (2b).

6.2.1 Getting the Latest Code

You can update your system using the ASUS. ASUS in turn uses CVS to access the **]po[** CVS code repository to get the latest code. The ASUS screen actually shows you the CVS command that it executes. Here is an example:

CVS Login (authenticates as the user "user" with password "password"):

```
export HOME=C:/ProjectOpen/projop
cvs -d :pserver:user:password@berlin.dnsalias.com:/home/cvsroot login 2>&1
```

CVS Update (gets the code):

```
export HOME=C:/ProjectOpen/projop
cd C:/ProjectOpen/projop/packages
cvs -z3 -d :pserver:user:password@berlin.dnsalias.com:/home/cvsroot update -d -P
-r v3-0-0-4-1 2>&1
```

The exact meaning of these commands is explained at http://www.nongnu.org/cvs/.

You can always use the user "anonymous" with an empty password to access the publicly available packages from **]po[**.

6.2.2 Getting the Latest Application Data

You can import the latest application data by restoring the last backup of the production server on your staging server. Please see further below for details on backup and recovery.

To load a backup “dump” into your database you can use:

```
psql -f backup_dump.sql
```

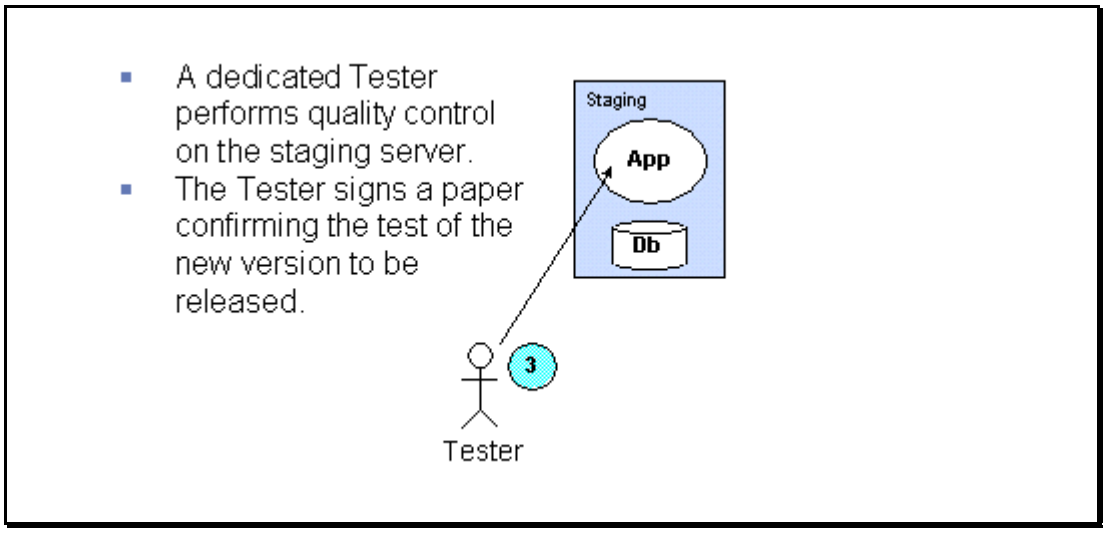
6.2.3 Updating the Data Model

Every new version of the]po[may require changes in the database in order for the application to work correctly. These database changes are managed using the “Advanced Package Manager” (APM). You can access the APM on the URL /acs-admin/apm/ on your server. Please click on “Install new Packages”, select all packages with status “update” and confirm.

ToDo: Explain manual sourcing of code if a database backup goes wrong.

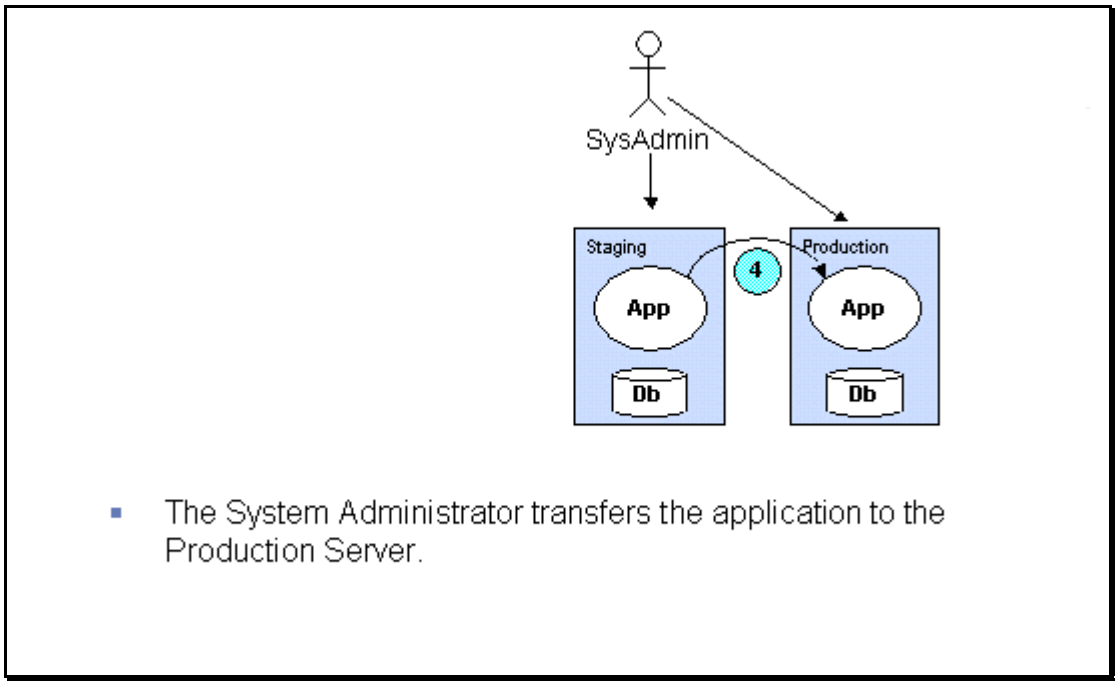
6.3 Testing

A “Tester” should verify that the application is running correctly before the staging process is repeated on the production server.



6.4 Productive Setting

“Productive Setting” is a repetition of the staging operation on the production server.



7 Helpdesk

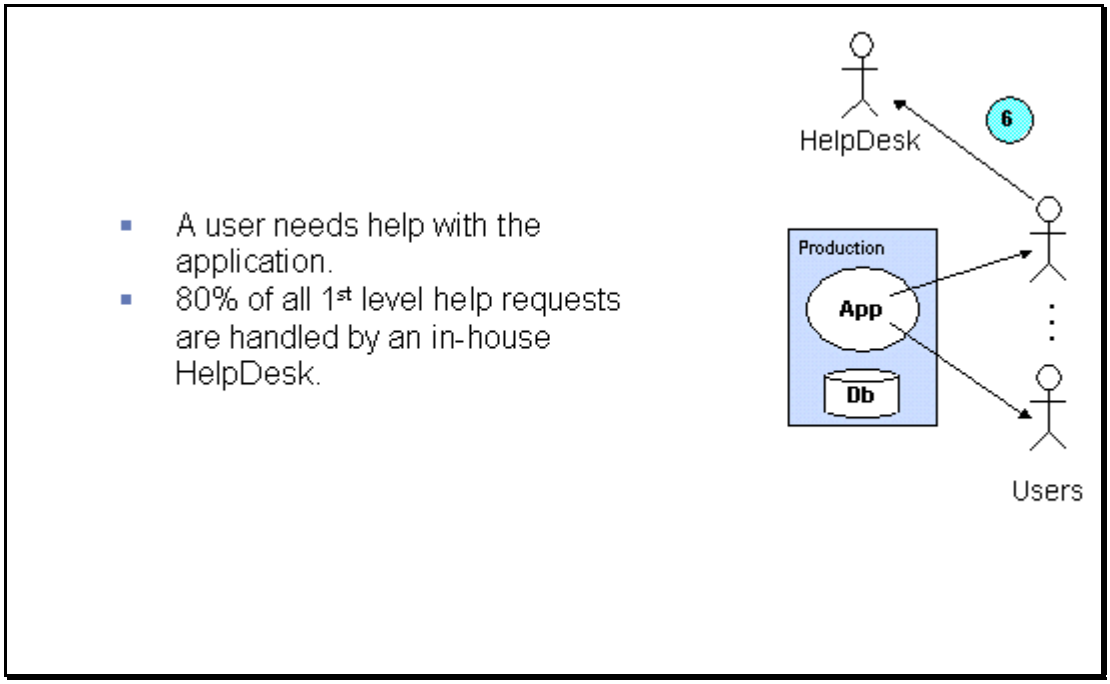
Helpdesk operations assure you that all of your users can use the system productively. In general you want to optimize the following parameters:

- Reaction time:
  User requests should be answered as quickly as possible in order not to waste time.
- Costs:
  You want to reduce the service costs of **]po[** or other help desk providers.

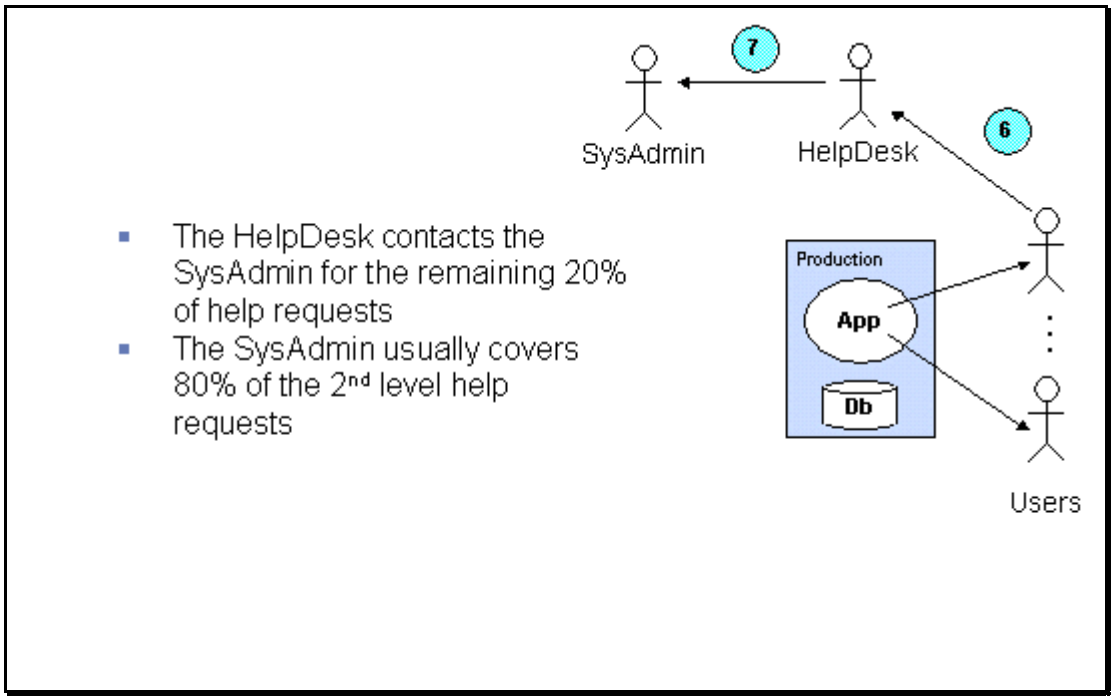
The best practice to optimize this reaction time / cost ratio is to use a staged system of:

- 1st level support (end-user support, typically in-house, dealing with questions & training issues),
- 2nd level support (support to the 1st level help desk, in-house our outsourced) and
- 3rd level support (support to your 2nd level support, typically outsourced).

7.1 1st Level Support

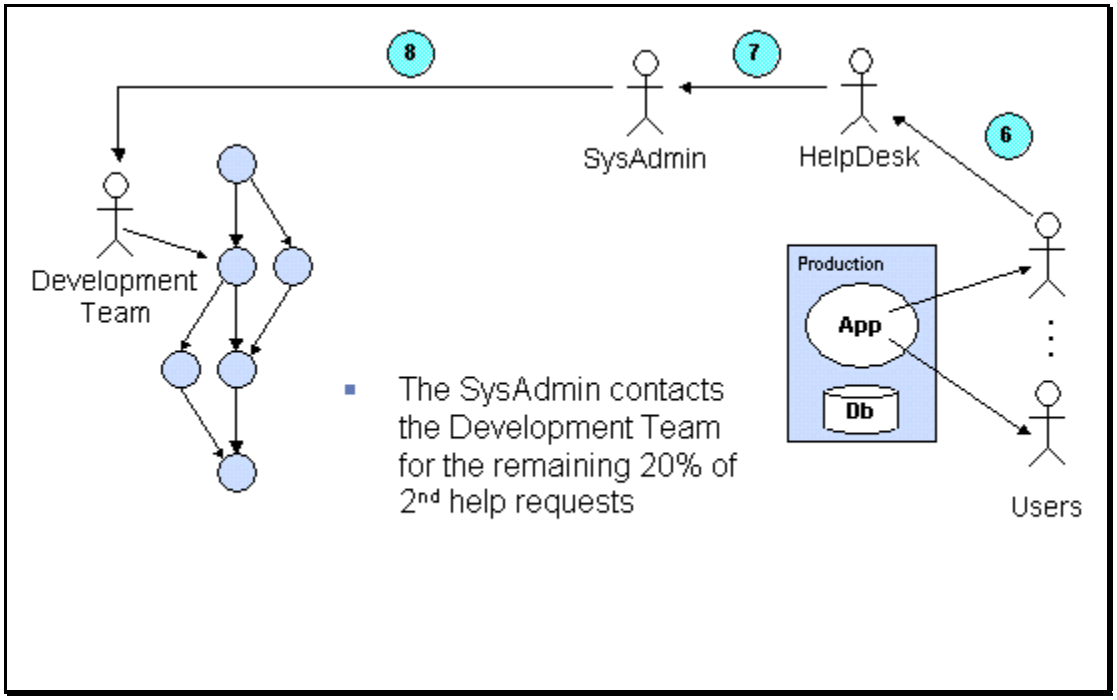


7.2 2nd Level Support



- The HelpDesk contacts the SysAdmin for the remaining 20% of help requests
- The SysAdmin usually covers 80% of the 2nd level help requests

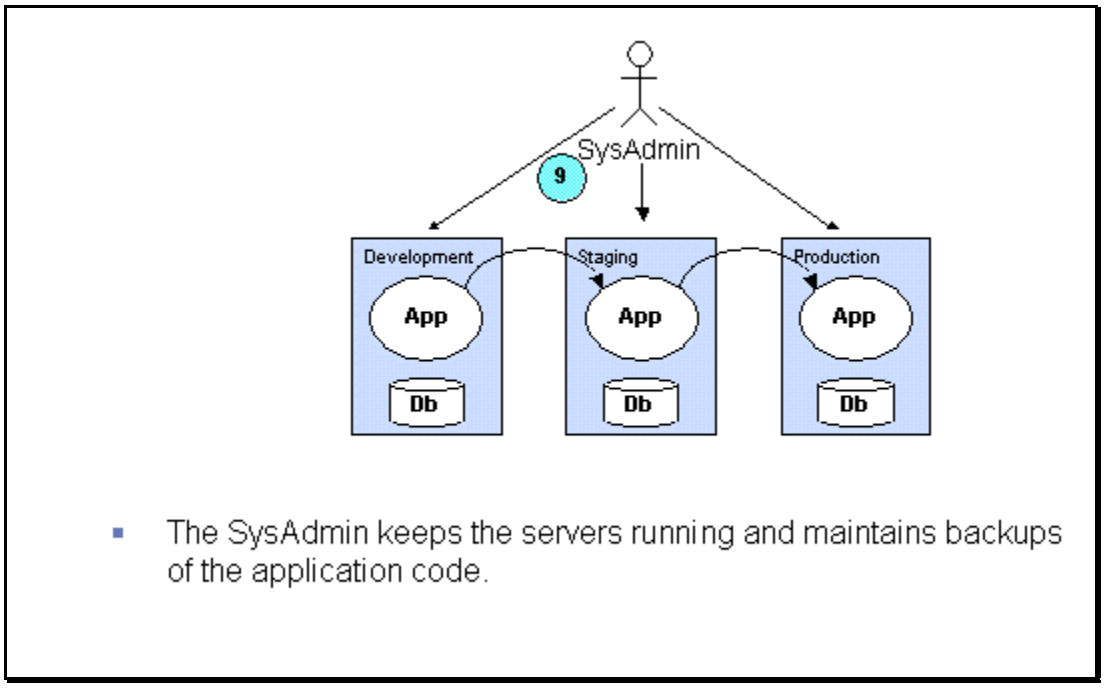
7.3 3rd Level Support



- The SysAdmin contacts the Development Team for the remaining 20% of 2nd help requests

8 System Administration

System administration includes all processes to keep the application running during the application lifecycle.



8.1 Application Code & Filestorage Backup

Application code (the content of the C:\ProjectOpen\projop\ folder) and filestorage (the content of the C:\ProjectOpen\filestorage\ folder or an equivalent location if you have changed the filestorage location) are plain code and can be backed up using a standard backup procedure such as:

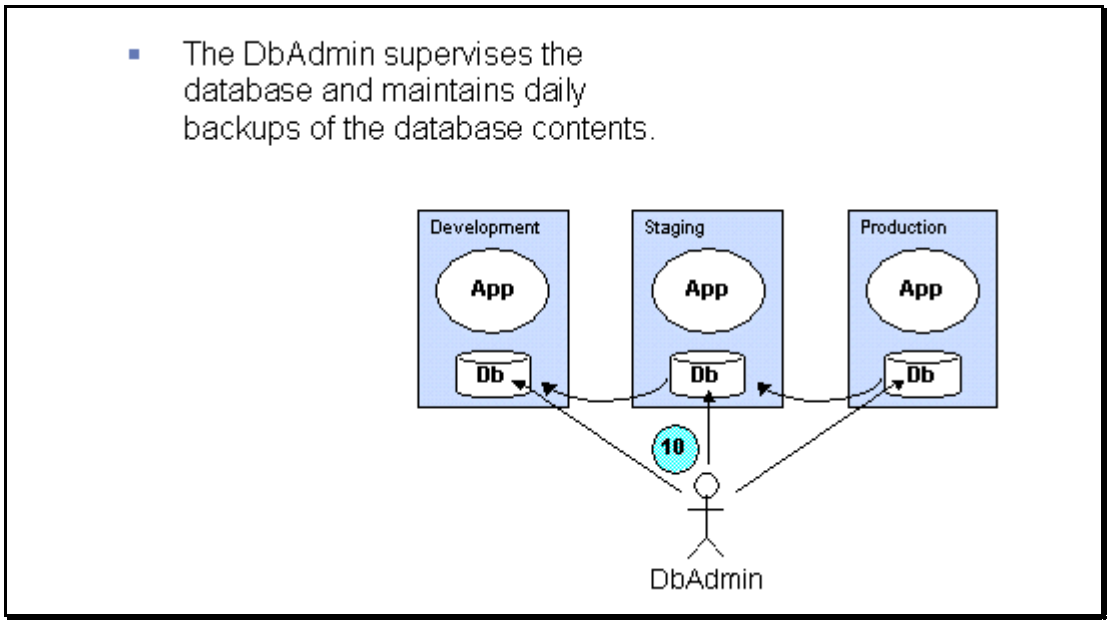
- Using Microsoft Backup (part of Microsoft Server operating system)
- Using an external hard disk
- Burning to a CD-Rom or DVD
- …

The other folders in the C:\ProjectOpen\ directory don't need to be backed up. You can reinstall them using the **]project-open[** Windows installer:

- cygwin
- doc
- nsd4
- pgAdmin
- preconf

All of these folders can be backed up during the operation of the system. You don't need to stop **]project-open[**.

8.2 PostgreSQL Database Backup



8.2.1 Built-In Full PostgreSQL Backup

]project-open[provides a built-in page to backup your PostgreSQL database. Just click on the "Admin" of your **]project-open[** installation and choose "PostgreSQL Backup". This page will backup your database into the C:\ProjectOpen\filestorage\backup\YYYYMMDD.HHMMSS\ folder by default (unless you change the location in the Admin / Parameters screen).

8.2.2 Interactive PostgreSQL Backup

]project-open[provides you with a script (Windows: Start -> Programs -> ProjectOpen -> Backup]project-open[Database") to backup your database. The daatbase dump "pg_dump.YYYYMMDD.HHMMSS.sql" is created in your "Documents and Settings" folder.

8.2.3 Manual Full PostgreSQL Backup

The Build-In Backup screen (see above) uses the PostgreSQL "pg_dump" command to backup the data. You can execute the command manually to achieve the same effect:

```
/usr/bin/pg_dump -c -O -F p -f pg_dump.YYYYMMDD.HHMMSS.sql
```

We recommend that you use the name "pg_dump" for the backup dumps, plus the current date in order to keep order in the backup dumps. The format "YYYYMMDD.HHMMSS" means (taking for example the 14th of August 2005, 7:50pm:

- YYYY the current year (2005)
- MM the current month (08)
- DD the current day (14)
- HH the current hour (19) in 24 hour format
- SS the current second (00)

8.2.4 Full PostgreSQL Backup Timing

You can execute the PostgreSQL backup during the execution of **]project-open[**, you don't need to stop the server. However, the backup will slow down the system to about 50% of its normal performance, so please choose some calm moments during the day.

We recommend companies with < 100 users to perform three full-backups per day, for example at 7am, 1pm and 10pm. However, these times can vary depending on your company profile. Also, you don't need to perform three backups per day.

8.2.5 Incremental PostgreSQL Backup

Incremental PostgreSQL backups are an option for large corporations (>1000 users).

Please contact us for more information or refer to the http://www.postgresql.org/ pages for more information.

8.2.6 Scheduling Automatic PostgreSQL Backups

We recommend that you schedule automatic PostgreSQL backups using the Windows "Scheduled Tasks" service. You can use the content of the "ProjectOpen-dbbackup.bat" file in your C:\ProjectOpen folder as an example.

8.3 PostgreSQL "Vacuum" Maintenance

PostgreSQL is very easy to maintain. The only maintenance measure is "vacuuming" the database in order to rearrange tables and to claim unused space. Your database will get slow if you don't vacuum it regularly.

8.3.1 Interactive "Vacuum"

]project-open[provides you with a script (Windows: Start -> Programs -> ProjectOpen -> Vacuum]project-open[Database") to vacuum your database.

8.3.2 Manual "Vacuum"

You can execute "vacuum" manually on the BASH command line:

```
/usr/bin/vacuumdb -f -a
```

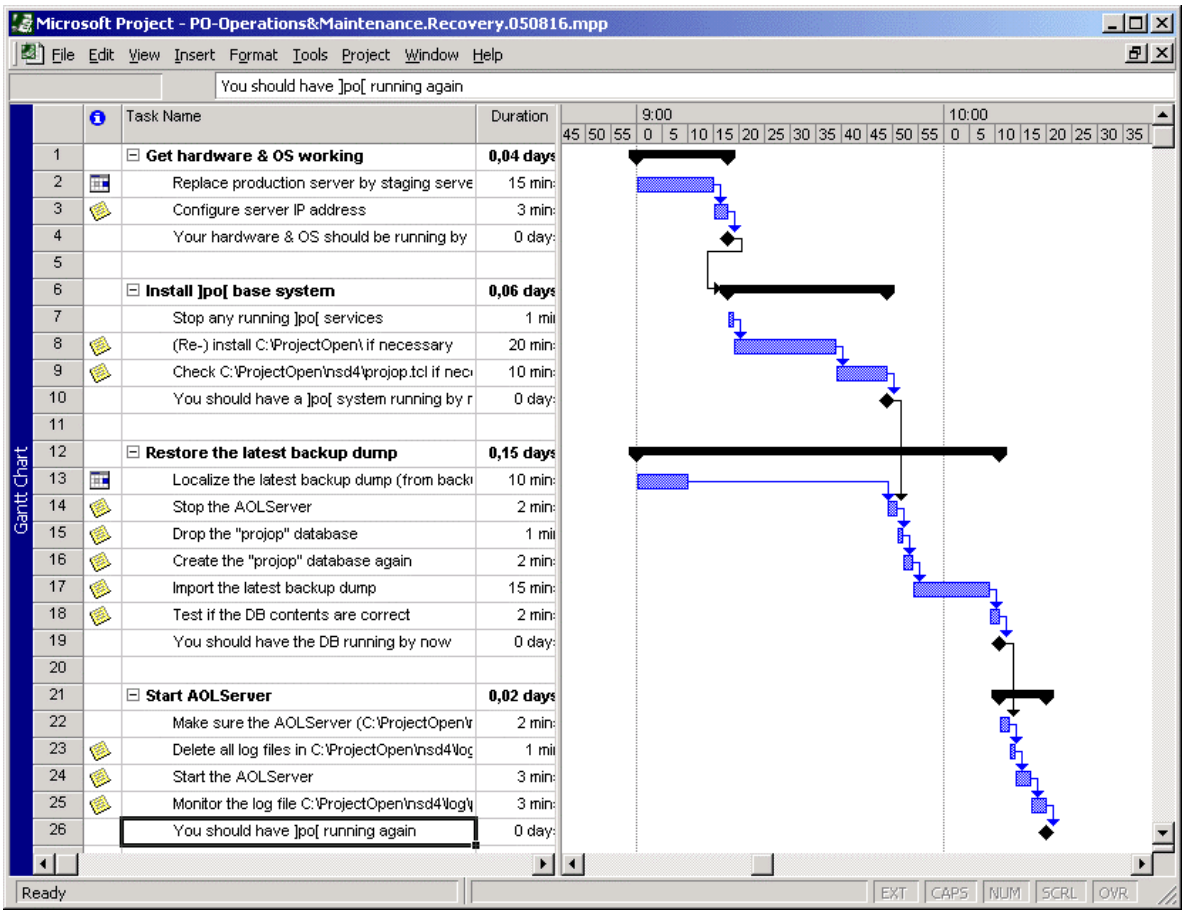
8.3.3 Scheduling Automatic "Vacuum"

We recommend that you schedule automatic PostgreSQL vacuum using the Windows "Scheduled Tasks" service. You can use the content of the "ProjectOpen-vacuum.bat" file in your C:\ProjectOpen folder as an example.

8.4 System Recovery

System recovery is the process of recovering a **]project-open[** system after system crash or another incident.

The following Gantt chart gives you an overview over the procedure. The recovery of a system should be possible within 90 minutes of time if backups have been made correctly and if there is spare server hardware available.



8.5 Restore PostgreSQL Database

This is the process of loading a backup dump into the PostgreSQL database.

8.5.1 Standard PostgreSQL Restore

The standard way to restore your PostgreSQL database is:

```
# psql projop -f pg_dump.XXXXXXXX.XXXXXX.sql
```

This command basically says: Take the pg_dump.sql backup file and execute all instructions in it.

This procedure works fine if there are no differences in the *structure* of the existing and the new database.

However, **this procedure will not work** if you have installed new modules or if you have otherwise added new tables or data structures to the PostgreSQL. This is the case in a recovery situation if you use the **]project-open[** installer to recreate a standard system and if your production system had a different package configuration.

In this case you will have to perform a “restore from scratch”:

8.5.2 Restoring PostgreSQL From Scratch

“Restoring from scratch” means that you delete your PostgreSQL contents completely and that you build them again from an empty database. The following steps walk you through the process.

Before restoring:

- Please make sure that the AOLServer has been stopped (Control Panel -> Administrative Tasks -> Services -> Stop the “AOLServer-projop” service)
- Please make sure that any pgAdmin III has been disconnected.
- Please make sure that the user “projop” is a PostgreSQL “super user” (i.e. can create new users). You can check this with the pgAdmin III application.

Drop the existing database:

```
# dropdb projop
```

This command delete all contents of the “projop” database and deletes all contents.

Create a new database:

```
# createdb projop –owner=projop
# createlang plpgsql projop
```

These commands create a new database with the same name. This new database is completely empty.

TSearch2 installation:

“TSearch2” is the full text database of PostgreSQL. This database gives some trouble during recovery, because its setup is not completely recorded in the backup dump. This is the reason why we haven’t included the full text search in the main **]project-open[** installation.

Does your application have TSearch2 enabled? Then please load the TSearch2 configuration into the database:

```
# psql projop –f /cygdrive/c/ProjectOpen/projop/packages/intranet-search-
pg/sql/postgresql/tsearch2.sql
```

Main recovery:

Now comes the main recovery part just like in the “standard restore”. This command will give lots of errors in the beginning, because the pg_dump.sql script contains code to drop the existing database tables and structures, that don’t exist in an empty database. So you can ignore these error messages.

```
# psql projop –f pg_dump.XXXXXXXX.XXXXXX.sql
```

After the recovery:

There are some minor issues with PostgreSQL versions < 7.2.4. These issues lead to a wrong ordering of the pg_dump.sql file and to errors in the recovery of several “views”. You can recreate these views using the following command:

```
# psql projop -f /cygdrive/c/ProjectOpen/preconf/poall.post-patch.sql
```

This is all.

Please see the “Diagnosing Errors” section if you have any kind of trouble.

Currently only TSearch2 can cause serious trouble. If so, please try to uninstall manually the “intranet-search-pg” data model by executing (psql projop –f …) the file “untsearch2.sql” and “intranet-search-pg-drop.sql” in the intranet-search-pg/sql/postgresql/.

8.6 Diagnosing Errors

If you should encounter any issues after a recovery please monitor the AOLServer log file C:\ProjectOpen\nsd4\log\projop.log. You can do this using your favorite text editor (and reload the contents from time to time) or you can use the following command that displays you the contents of the file as it grows:

```
# tail -f /cygdrive/c/ProjectOpen/nsd4/log/projop.log
```

Ronda Sant Antoní, 51 1° 2a
08011 Barcelonaa, Spain
Tel.: +34 93 325 0914
Fax.: +34 93 289 0729