ProcessManager Technical Introduction

Table of Contents

Introduction

Audience

The intended audience is members of IT departments maintaining an installation of ProcessManager.

Purpose

The purpose of this document is to explain the overall workings of the ProcessManager installation, as well as how the different components interact and highlight problems what might arise with them, especially when changing servers doing and other architectural changes at the site.

ProcessManager

What is ProcessManager?

ProcessManager is a tool to aid in planning and visualize processes and workflows in a company. It mainly does so by helping the user divide an overall process into smaller sub-processes, and attach documents (i.e. webpages, Word documents or pictures, or even other processes) to them. To help with the visualization, it is possible to both use shapes and colors as visual hints.

How ProcessManager works

The main user interface runs in Adobe Flash, in the employees' browsers. The Flash application communicates with a PHP website, either hosted by Process-Manager ApS, or at the customer's own site. The PHP website has been tested with Apache and IIS (although any web server capable of running a recent version of PHP should be usable).

Data from the application itself is stored in a database (currently, MS SQL server and MySQL are supported), and documents uploaded to the ProcessManager service is stored on the file system.

User authentication and authorization is either done by an internal ProcessManager system, or by Windows Authentication/Active Directory.

Components

The frontend

The Flash frontend should run on any browser capable of executing Adobe Flash programs. That being said, there is a number of gotchas to look out for, especially when running in Internet Explorer 8+ and opening Office documents:

- The ProcessManager site must either be in Internet Explorer's *Trusted Sites* or *Local Intranet* zone.
- *File Downloads* must be enabled for that zone.
- *Include local directory path when uploading files to a server* must be enabled for that zone[1]

---

[1] This is only needed if you want to link to local network shares.

Another problem might happen if the Flash application is run in low quality mode (as e.g. Citrix XenApp/MetaFrame might do automatically): Some window titles and labels might not get updated properly. This seems to be a problem with the Adobe Flex product. It is recommended never to run ProcessManager in anything but high quality mode. As there are almost no animations, this should not be a problem unless running in a very low bandwidth area.

Backend webserver

PHP is used for running the backend. The PHP installation must be rather standard, and using settings such as *safe_mode* (removed from PHP as of 5.4.0) and *magic_quotes_gpc* is highly discouraged. Please ask Process-Manager ApS is you have any specific special settings needed.

On Apache, PHP must be installed as mod_php. CGI mode is not supported.

On IIS, PHP must be running under FastCGI. ASAPI mode is not supported by PHP anymore. Unfortunately, there is no 64-bit version of PHP. This presents some challenges when dealing with the database in a 64-bit Windows installation, but besides that, there are no adverse effects.

Database

On 64-bit Windows systems, no matter which DB engine has been chosen, all ODBC configuration must be done through the 32-bit ODBC manager, *odbcad32.exe*.

For MySQL, in most configurations, ProcessManager should be able to configure everything automatically. However, InnoDB must not be disabled, as the ACID features of that database engine is needed.

For MS SQL, there are some things to consider if moving to another SQL server after installation:

- *SQL Authentication* must be enabled.
- A user and database must be created for the system, and the user must have the correct rights for the database: *db_datareader*, *db_datawriter*, *db_ddladmin* and *db_owner*.

Authentication and authorization

The authentication system in ProcessManager supports both coarse access controls (user roles), and more fine grained control to read and edit parts of the process tree. It can either be managed internally by ProcessManager itself, or in cooperation with Windows Authentication.

Internal authentication system

The internal authentication system runs entirely inside the database. It uses a simple popup for entering user credentials. This is the only supported authentication system if the ProcessManager system is hosted by Process-Manager ApS.

Windows Authentication/Active Directory

Windows Authentication is implemented in ProcessManager using the file system rights to check if the user has a certain role: If the file is readable, the user has that role. In the ProcessManager installation, three files are inside the directory *roles*:

- *Viewer* determines if the user has access to ProcessManager at all.
- *Author* determines if the user has write access.
- *Admin* determines if the user has administration access.

When backing up and restoring the ProcessManager installation directory, remember to keep the security settings of the roles subfolder intact.

For the fine-grained access, the internal ProcessManager authentication system is still used: It just gets users from Windows Authentication, and adds them to the internal access control database. This may present a challenge if updates of the AD database changes user's domain changes. However, it is rather straightforward to do a search and replace in the column *username* in the table *users*.

When using Windows authentication, it is important not to use the normal *temp* folder for uploading files, as PHP will not be able to change rights after moving the file to the final destination, meaning that only the user who uploaded the file will be able to access it. See the installation manual for further details.