ProcessMaker Installation & Upgrade

Installation

Necessary resources

- How many users/cases per day?
- 1 CPU core per 4 GB of memory. 128 MB per user, allow for 30% spikes.
- **Redundancy:** The cluster should support one server being unavailable at peak.
- Server sizing.

Supported stacks

- PM has been tested on CentOS and Windows Server. While it works in Debian/Ubuntu, it has not been extensively tested by ProcessMaker's QA team.
- Supported stacks.

Weird Expiration Data Error

- In ProcessMaker 3.4.5 and later, the Admin user expiration date calculates as one year after starting from the ProcessMaker installation or workspace creation.
- It may become necessary to update when the admin user account expires.
- In ProcessMaker 3.4.4 and earlier versions, manually modify the date in the database using PhpMyAdmin or the MySQL console.
- **Example:**

  ```
  UPDATE RBAC_USERS SET USR_DUE_DATE='2030-12-12' WHERE USR_UID = '00000000000000000000000000000001'
  ```

ProcessMaker Upgrade (to 3.3)

General suggestions

- All changes in a development environment first!
- Do NOT delete the old ProcessMaker directory and then copy the new ProcessMaker directory to the same place. You will lose:
    - Essential configuration information.
    - Uploaded input documents.
    - Attached files.
- Instead, decompress the file to overwrite any existing files without affecting any other files.

General suggestions (cont.)

Roadmap

- Version 3.2 (or even 3.0x / 3.1x)
- Version 3.2.2 or 3.2.3
- **Version 3.3**
- Version 3.4

Upgrade Schedule Template

1. Deploy new environment with a supported OS.
2. Install supported stack.
    - Apache/NGINX
    - PHP
    - MySQL
3. Create backup.
4. Database backup creation.
5. Additional setup if needed.
6. Standard upgrade procedure.
7. Basic functionality testing.
8. Specific functional tests.

Backups

- Make **full backups**

1. Make an archive of the ProcessMaker installation directory using the tar command, which is located at `/opt/processmaker`:
   ```
   tar -czf BACKUP-FILE.tar.gz /opt/processmaker
   ```
2. Make a backup copy of the entire MySQL database using the `mysqldump` command:
   ```
   mysqldump -u root -p -F -x --databases wf_workflow > BACKUP-FILE.sql
   ```

File permissions

- Uncompress and overwrite the files in the `processmaker` folder with their new version, keeping the permissions, according to your system.
- Change the owner of the `processmaker` files to the user who runs Apache or NGINX.

MySQL compatibility

- To avoid problems with triggers on MySQL 5.5 and greater, change logging by rows instead of by statements (default).
- This can be done on the `my.cnf` or `my.ini` file.
- `binlog_format = row`

Workspace upgrade

- Run the `processmaker upgrade` command to make all the workspaces run on the new version.
- Install `php-cli` if need, or use the full path if it is installed (may not appear in `$PATH`).

Apache

- ProcessMaker 3.3 has a different VirtualHost configuration in Apache than in version 2.8.X and earlier.
- The upgrade process does NOT update the Virtual Host configuration automatically.
- It is necessary to change the Virtual Host configuration for ProcessMaker manually.
- Update the `pmos.conf` file and then restart the Apache service.

Database upgrade

- After upgrading the ProcessMaker source code, the MySQL database(s) used by ProcessMaker need to be upgraded as well.
- Database upgrade: `./processmaker database-upgrade workflow`.
- Flush cache: `./processmaker flush-cache`.
- It is recommended also to completely clear your browser cache (or use incognito mode) when upgrading.

Migrating the list of cases

- To move the cases to their corresponding tables, execute the processmaker migrate-new-cases-lists command. (Enterprise edition only).

If something goes wrong...

- Log in as `root` and remove the ProcessMaker installation directory and replace it with the backup:

  ```
  rm -fr /opt/processmaker
  ```

  ```
  tar -C /opt -xzf BACKUP-FILE.tar.gz
  ```

- Then, restore the databases:

  ```
  mysql -u root -p < BACKUP-FILE.sql
  ```

Backups

- It is recommended to do periodic backups.
- Backups only work on the same platform (Windows -> Windows and Linux -> Linux) because the table names in MySQL are not case sensitive in Windows, which can create problems.
- It is possible (but not ideal) to backup from older to newer versions.

Automatic backups

- In Linux/UNIX, backups can be scheduled as a cron job.
- 12-hour backup of `workflow` workspace:
  `* */12 * * * root /opt/processmaker/processmaker workspace-backup workflow`
- Even better, have 12-hour backups with a timestamp to roll back to states older than 12 hours:
  `* */12 * * * root /opt/processmaker/processmaker workspace-backup workflow /root/backups/workflow_`date +%F_%H-%M`.tar`
  `* */23 * * * root tmpwatch 720 /root/backups`