PowerCard Release Guidelines

Generic BitBucket release guidelines:

* HPS could create a repository in BitBucket when necessary using the “master” branch;
* SIA creates the “hps” branch after first commit for each repository, so that since the second commit HPS must commit in this branch (other branches will be unavailable to HPS);
* Each patch/release released into BitBucket must be tagged using a distinct incremental TAG (adherent to FTP release, in example “PATCH022”);

Software source code release guidelines:

* All source code must be released through BitBucket;
* Report project must be release in BitBucket too (.java and .jasper files both);
* Customized tools necessary to compilation should be released in BitBucket too;
* Scripts necessary to be run by Jenkins jobs should be released in BitBucket too;
* SIA creates its Jenkins jobs following the provided “PowerCARD Jenkins Configuration” document (this document must include C++ and PowerCardConnectAPI compilation jobs, and should be updated when compilation jobs change);
* Each change in a software source code implies a version increment of the compiled software (for maven projects the version must be stored in the pom.xml file, for C++ softwares and reports it is desirable to have a version too);
* Maven dependencies must not be released into BitBucket, but HPS must provide (through FTP Server) every dependency not present in the official Maven repositories, in order to let SIA to update its Maven repository for the compilation;
* C++ dependencies could be released into BitBucket and updated when necessary (right now, SIA doesn’t enough information about C++ compilation, so this part can be discussed later when C++ build process knowledge will be provided);

DB release guidelines:

* All scripts regarding DB structure must be released through BitBucket (scripts must be applicable to Production environment ensuring data consistency);
* Scripts necessary to launch DB update process should be released in BitBucket too;
* Differently from software source code, each DB patch must be included in a different subfolder in order to apply to the DB only the increment. (A naming convention should be applied, in example the name of the folder could be “PATCH022” aligned to the BitBucket TAG described before)

Still via FTP server:

* Backup Scripts (?);
* Documents (release note, more compilation and installation guides when necessary);
* New Maven dependencies used during maven project compilation not present in official Maven repositories
* A component traceability file that list all updated artifacts in the current release and relative version if available. For example it can contains:

|  |  |
| --- | --- |
| Client | 3.31 |
| PowerCardAPI | 1.0.2.GA |
| Report | 2.4 |
| CIS | 5.31 |

To avoid:

* Source released via FTP Server (see PowerCardAPI in MPatch001 release);
* Different source code repositories for the same artifact;
* Different DB updates in a single release (see PowercardAPI in MPatch001 release);
* Configuration data necessary for compilation (DB URL and credentials in example) wired inside source code or pom.xml file;

Temporary management:

* Compiled artifacts will still deployed through FTP Server until SIA build processes will be correctly up;

Example of BitBucket Repositories for PowerCard product:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Web | Client | … |  |  |
| Server | … |  |  |
| PowerCardApi | … |  |  |
| Tools | Generator | … |  |  |
| Librairies | … |  |  |
| Plugins | Maven-flex2-plugin | … |  |
| Tests | … |  |  |
| PowerCardApi | … |  |  |
| Scripts | Prepare\_env | Compile\_librairy | LibrairyCompile.jar |  |
| \*.sh |  |
| \*.jar, … |  |  |
| \*.sh |  |  |  |
| sqlj | bin | JCedic,… |  |  |
| demo | jpub | \*.sql |  |
| lib | runtime12.jar |  |  |
| … |  |  |
| fb3 | ClientLibrairy | ALBARAKA\_INT | … |  |
| … | … |  |
| Librairy | as3crypto | … |  |
| … | … |  |
| sdks | 3.6.0-v3-commun | … |  |
| report | PowerCardReportApi | src | \*.java |  |
| jasper | \*.jasper |  |  |
| \*.grf |  |  |
| Unix | CIS | src | \*.cpp |  |
| HOSTBANK | src | \*.cpp |  |
| lib | \*.so |  |  |
| shl | \*.sh |  |  |
| DB | Patch021 | BO | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| FE | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| Patch022 | BO | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| FE | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| MPatch001 | BO | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| FE | Dev\_DB\_Repository | … |
| logsDB |  |
| MainPatchDBSetup.sh |  |
| \*.sql |  |
| … |  |  |  |