

 Retour au Dossier Organisationnel	
 Sujets connexes	Cartographie fonctionnelle de LoGeAs Procédure de versionning des codes sources (procédure #30) Procédure de prise en charge d'un ticket - ProjeQtOr (procédure #04) Procédures de gestion du dépôt GitLab (procédure #??) Procédure de dépôt des codes sources (procédure #09)

Procédure de gestion des versions de LoGeAs (procédure #06)

Informations qualité

Suivi des modifications majeures	5 mars 2015 - Nicolas Marchand - Création du document 1 décembre 2015 - Nicolas Marchand - Modification par rapport aux tests à passer suivant le niveau des mises à jour 14 décembre 2015 - Guillaume Natali - Modification des règles de numérotation 02 aout 2017 - Nicolas Marchand - Portage sur DoKuWiKi & Evolutions 28 août 2020 - Nicolas Marchand - Evolution raisons changement de version Janvier 2026 - Nicolas Marchand - Refonte du document suite à l'automatisation du versionning globale
Suivi des approbations	Cartographie fonctionnelle
Objet	L'objet de ce document est de définir les différents types des versions mises à disposition, les différences entre elles, et les actions à faire lors de la sortie d'une nouvelle version.
Destinataires	- Validation des modifications : Gérant - Approbation du document : Equipe dev & Equipe Ass

Objectif du versionnage

Selon le [Grand dictionnaire terminologique](#), le versionnage (équivalent francophone de l'anglais *versioning*) est le mécanisme qui consiste à conserver la version d'une entité logicielle quelconque, de façon à pouvoir la retrouver facilement, même après l'apparition et la mise en place de versions plus récentes [Wikipedia](#)

Définition des types des versions de LoGeAs

Pour LoGeAs, une **version du logiciel** correspond à un état donné de l'évolution du produit mis à disposition des clients à jour de leur contrat. Contrairement à certains éditeurs qui différencient le numéro de version de développement de celui de commercialisation (Microsoft par exemple), Logeas Informatique utilise une numérotation unique

La numérotation utilisée est une série de 4 chiffres séparés de points (LoGeAs v11.0.3.1). Pour permettre une meilleure compréhension, nous nommerons cette suite de chiffres de la manière suivante.

		Logeas Informatique considère le numéro "Majeur" plus comme un numéro "marketing" que "technique", bien que parfois les deux coïncident. <u>Par exemple :</u> * la version 5.0 a vu la migration de la base de données du logiciel de BDE (propriétaire Borland) vers un système multi-base compatible SQL (par exemple SQLite ou Postgress) * la version 6.0 est une simple évolution de la version 5.0, sans changement majeur, mais a été mise en place pour les raisons précédentes après 2 ans et demi de bons services de la V5.... * la version 11.0 à vu le début de l'interface full web
11	Majeur	Le numéro de version « Mineur » est changé quand la version met en place de nouvelles fonctionnalités et/ou des changements sont réalisés dans la structure de la base de données (ajout d'un champ par exemple). Quand il y a des changements de règlementation pris en compte ou une nouvelle certification
0	Mineur	Le numéro de version « Release » est changé à chaque évolution du logiciel (plus exactement à chaque évolution d'une composante du logiciel (voir plus loin).
0	Release	Le numéro de version «Build» est utilisé en interne pour le suivi des exécutables, des bugs et des tests.
0	Build	

Règles

1. Une fois qu'une version est publiée, le contenu de sa version NE DOIT PAS être modifié. Toute modification DOIT être publiée dans une nouvelle version.
2. Quand le segment Majeur est changé, les "Mineur", "Release" et "Build" sont remis à 0. De même quand le segment "Mineur" est changé, les "Release" et "Build" sont remis à 0 et ainsi de suite.

Allons un peu plus loin

LoGeAs n'est pas un logiciel monolithique en ce sens qu'il s'appuie sur une série de "sous-logiciel" qui constitue la **galaxie LoGeAs**.

Si on va plus long la galaxie se compose de

Serveurs	LogeasWeb	C'est le serveur qui donne accès aux base de nos clients. Il s'agit principalement d'un exécutable windows, écrit en Delphi
	PGI	C'est le serveur qui gère la base de données de nos utilisateurs. Il s'agit aussi d'un exécutable Windows, écrit en Delphi
	Nono	C'est un serveur annexe qui gère une série de base de données anonymisées, a usage technique. Il s'agit d'un exécutable Windows, écrit en Lazarus

Interfaces	LoGeAs (ou client lourd)	Il s'agit de l'interface "historique" qui est actuellement en cours de portage vers angular. Elle est écrite en Delphi et génère un exécutable windows qui est installé chez le client
	LoGeAs.fr (ou client full-web)	Il s'agit de l'interface utilisateurs, qui remplacera à terme le "client lourd". Elle est écrite en Angular, et est utilisable via un navigateur Internet
	Test.LoGeAs.fr	Il s'agit de l'interface de test final de l'interface précédente
	Assistance	Il s'agit de l'interface utilisé uniquement par l'assistance pour la gestion, les ticketing... Elle est aussi écrite en Angular
	Cartographie fonctionnelle	Il s'agit d'une interface dédié à la gestion de la qualité au service de nos certifications NF. Ecrite en Angular
	Stat-Union	Il s'agit d'une interface dédié à l'ensemble de nos client EPUDF est qui dédié à la consolidation des comptes à vison statistique

- Chaque sous ensemble de la galaxie à **son propre numéro de version** qui évolue en fonction des évolutions, chaque numéros respecte les règles énoncées plus haut
- Les **digits Majeur et Mineur** sont commun à toutes les interfaces. Si un élément de la galaxie doit changer un de ces digits, la prochaine mise à jour des autres éléments doit le prendre en compte
- Le versionning de chaque élément est géré dans son outil de développement et repris au niveau de GIT.

Git ne travaille qu'avec ces numéros.

Définition et évolution de la version globale

Dans ce contexte, il est nécessaire de définir une **version "globale"** du logiciel, afin de pouvoir savoir dans le suivi qualité sur quelle version on travail, on test... et surtout qu'elle réalité cela recoupe.

Règle d'évolution de la version globale

Afin de respecter globalement les règles de gestion des numéros de version, les règles suivantes sont appliquée pour "calculer" le numéro de version globales.

Majeur	Le numéro "majeur" est aligné sur le numéro majeur le plus élevé des sous composants. Si le numéro majeur est incrémenté par rapport à la version précédente alors les autres digit sont remis à zéro
Mineur	Le numéro "mineur" est aligné sur le numéro mineur le plus élevé des sous composants. Si le numéro mineur est incrémenté par rapport à la version précédente alors les autres sous-digit sont remis à zéro
Release	Le numéro "release" est incrémenté à chaque fois qu'une modification est faite dans un sous composant
Build	Non utilisé à ce stade

* C'est la version globale qui est utilisé dans la partie qualité, assistance...

Changement de version et actions à faire A REPRENDRE

Dans le cas d'une « Majeure »

Avant le début des développements

- Séparation de la documentation utilisateurs wiki (accès concurrent) [\[PROC-NFlog-6\]](#)

Avant de « livrer » la version

- Validation de tous les tests définis dans ProjeQtOr [\[PROC-NFlog-10\]](#)
- Evolution et Archivage de la documentation technique [\[PROC-NFlog-6\]](#)
- Mise en ligne sur le site de la description des modifications effectuées à partir du log de SVN (actualité sur le site) [\[PROC-NFlog-7\]](#)
- Taguage de l'arborescence SVN avec le numéro de version et enregistrement des binaires [\[PROC-NFlog-7\]](#)
- Archivage des fichiers binaires, Installateur et Mise à jour dans le dossier Cloud\LoGeAs Historique des versions [\[PROC-NFlog-7\]](#)
- Changement de l'URL des mises à jour et du dossier sur le serveur [\[PROC-NFlog-7\]](#)
- Mise à jour de la documentation EPUDF REGALE [\[PROC-NFlog-6\]](#)

Après « livraison » de la version

- Envoi par mail à tous les utilisateurs de l'information
- Dépôt des codes, documentation ... auprès de l'EPUDF [\[PROC-NFlog-8\]](#)
- Dépôt des codes et des binaires dans le coffre numérique [\[PROC-NFlog-8\]](#)

Dans le cas d'une « Mineure »

Avant de « livrer » la version

- Validation de tous les tests définis dans ProjeQtOr [\[PROC-NFlog-10\]](#)
- Evolution et Archivage de la documentation utilisateur [\[PROC-NFlog-6\]](#)
- Evolution et Archivage de la documentation technique [\[PROC-NFlog-6\]](#)
- Mise en ligne sur le site de la description des modifications effectuées à partir du log de SVN (actualité sur le site) [\[PROC-NFlog-7\]](#)
- Taguage de l'arborescence SVN avec le numéro de version et enregistrement des binaires [\[PROC-NFlog-7\]](#)
- Archivage des fichiers binaires, Installateur et Mise à jour dans le dossier Cloud\LoGeAs Historique des versions [\[PROC-NFlog-7\]](#)
- Mise à jour de la documentation EPUDF REGALE [\[PROC-NFlog-6\]](#)

Après « livraison » de la version

- Envoi d'un mail à tous les utilisateurs pour leur indiquer la mise à jour
- Dépôt des codes et des binaires dans le coffre numérique [PROC-NFlog-8]
- Extension de la certification du logiciel

Dans le cas d'une « Release »

Avant de « livrer » la version

- Validation des tests en rapport avec les fonctionnalités impactées [PROC-NFlog-10]
- Evolution et Archivage de la documentation technique [PROC-NFlog-6]
- Mise en ligne sur le site de la description des modifications effectuées à partir du log de SVN (actualité sur le site) [PROC-NFlog-7]
- Taguage de l'arborescence SVN avec le numéro de version et enregistrement des binaires [PROC-NFlog-7]
- Archivage des fichiers binaires, Installateur et Mise à jour dans le dossier Cloud\LoGeAs Historique des versions [PROC-NFlog-7]
- Mise à jour de la documentation EPUDF REGALE [PROC-NFlog-6]

Après « livraison » de la version

- Dépôt des codes et des binaires dans le coffre numérique [PROC-NFlog-8]

From:
<https://wiki-logeas.fr/certif/> - **dokewiki-certif**



Permanent link:
<https://wiki-logeas.fr/certif/doku.php?id=certif:procedure:develop:gestionversion>

Last update: **2025/12/30 14:23**