Titan Software Release Procedures
Version 1.3

Overview

This document describes the Titan procedures for the release of software project deliverables. These procedures are designed to ensure a smooth, orderly transition of products from Titan to its customers, while maintaining a clear release history along with source- and version control (source control and versioning are described below).

In general, software development efforts are organized into projects that produce new and/or enhanced application programs. Application programs may be released at any time within a software project. Releases consist of:

1. An application installation program, if applicable.
2. The executable application itself (.exe or .dll).
3. Application release notes.
4. A user's manual, if applicable.
5. Application source code, if applicable. When the customer has purchased source code, Titan delivers all source necessary for a third party developer to rebuild the delivered executables, given the same set of developmental tools as used by Titan.
6. Database schema, or database alter script, if applicable.

Software Source Control and Versioning

To maintain software application release control, and to ensure a consistent maintenance and upgrade capability, all software developed and released by Titan is subject to source control and versioning.

Source Control

Currently Titan controls source code using branching and labeling as supported by Microsoft SourceSafe. A branch is used as a separate path to a deliverable on a common project. A new branch is required when a previous release of a delivered product requires modification prior to release or a new version. Branch points enable Titan to continue with future software deliverable while at the same time maintaining on-going support of a customer's current software product release.

A *label* is used to denote a particular build. When a product is ready for test, the developer updates the version number, builds the target executable or dll, checks in all applicable source code to SourceSafe, and creates a new SourceSafe label using a form of the current version number (e.g. "Version 2.0.8"). The executable is transferred to the testing department along with the label information. If a bug is found, the tester issues a software modification request (SMR), the code is reworked, and when the new version is ready to be retested, a new label is created in SourceSafe. Once the code has passed all tests, the executable is transferred to the release manager, along with the SourceSafe label information. The release manager checks out the version of the source code from SourceSafe indicated by the release version number and includes it in the release package.

Versioning

When a software product is released to the customer, a version number is associated with the release, consisting of three fields as described in the table below.

Field	Field type	Definition
x	*project_number*	Number corresponding to the project (e.g. “2” for Batch Editor).
y	*release_number*	Current release of the project. This number is incremented as a project’s requirements change over time.
z	*build_number*	This number acts as a build number, and is incremented each time the code enters the test phase.

Table 1. The Titan *x.y.z* form of software versioning

The project number is incremented when there are major changes to the application, involving both its functionality and database structure. These major changes are usually associated with a particular new project or project phase.
The release number is incremented with each customer release, and may represent bug fixes, new functionality, or both. If the customer has version 2.1.z, the next release the customer should receive – regardless of number of bug fixes or enhancements contained in it – should be version 2.2.z.
The build number represents the internal release or test iteration, and is incremented each time a new build is sent to the testing department.