

Release Management

TwoGo Codelines

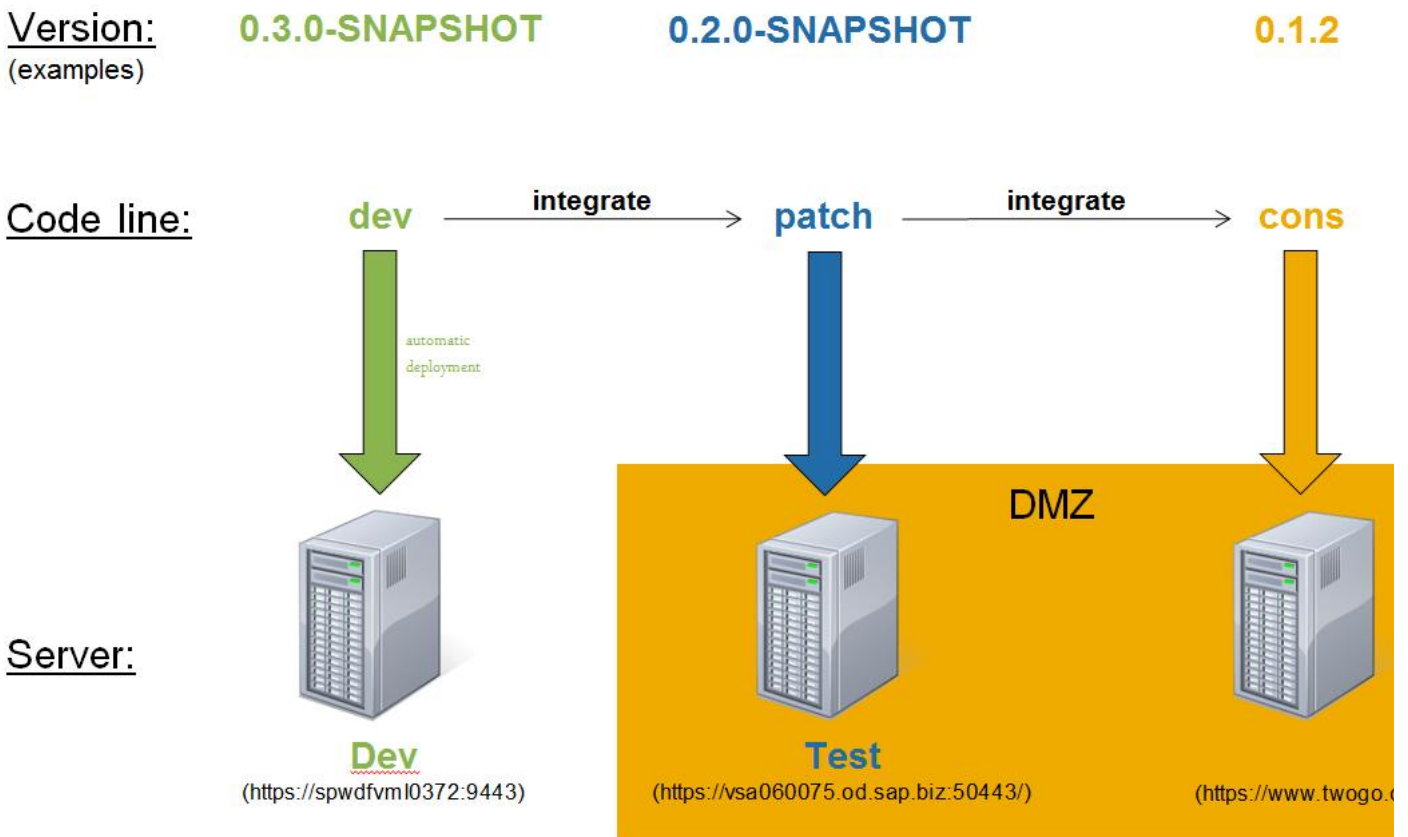
TwoGo source code is located in Perforce Port **performance3406:3406**, depot **ptg**. The depot contains 4 (standard SAP) code lines: **dev**, **export**, **patch** and **cons**, whereas TwoGo only uses 3 of them: **dev**, **patch** and **cons** (and to be precise, TwoGo uses them not in their intended SAP functions).

- **dev** code line is owned by all TwoGo developers. No restriction applies to any check-ins, except the code must be buildable on CI Server (Jenkins: <http://spwdfvm1246.wdf.sap.corp:8080/jenkins/>). After each check-in an automatic deployment (incl. new change list) is triggered from CI Server to central **Dev machine**.
- **patch** code line is owned by TwoGo Quality Management (QM) Team. This code line is created by integrating dev code line to patch code line at a certain point in time (currently: every second week). TwoGo developers have full access (read/write) to this code line, however they're only allowed to change file(s) in patch code line on QM Team request. Deployments from patch code line will take place on request to central **Test server**.
- **cons** code line is owned by TwoGo Administrators. This code line is created by integrating patch code line to cons code line at a certain point in time (currently: every second week before dev gets integrated to patch). TwoGo developers have full access (read/write) to this code line, however they're only allowed to change file(s) in cons code line on Product Owner request (Hot Fix). Deployments from cons code line will take place on request to central TwoGo productive server (<https://www.twogo.com>).

TwoGo Versioning

TwoGo version numbers are formatted as proposed by Apache Maven project
 <major version>.<minor version>.<incremental version>-<qualifier>

- TwoGo increases major version numbers to indicate incompatible API changes and/or new/reworked major features (incl. bug fixes)
- TwoGo increases minor version numbers to indicate compatible API changes and/or new/reworked features (incl. bug fixes)
- TwoGo increases incremental version number to indicate bug fix provisioning
- TwoGo uses Apache Maven qualifier **SNAPSHOT** to indicate instable (frequently changed) versions, which have not been officially released



1. After integrating patch code line to cons code line, the qualifier **SNAPSHOT** is removed from version number (in cons code line) to release (via **TwoGo Nexus**) the version.
2. In case a severe bug fix (a.k.a. HotFix) must be applied to productive server, the fix is checked in to cons code line, build on CI server and deployed to productive server. To identify this fix, TwoGo increments the incremental version number (e.g. 0.1.1 → 0.1.2)
3. After integrating dev code line to patch code line, the minor version number gets incremented (in dev code line) (e.g. 0.2.0-SNAPSHOT → 0.3.0-SNAPSHOT)

In case TwoGo plans to release a new major version, the major version number gets incremented (e.g. 0.4.0-SNAPSHOT → 1.0.0-SNAPSHOT). It is considered a best practice to reset all subsequent version numbers to 0, when incrementing a preceding version number. (e.g. 1.2.0-SNAPSHOT → 2.0.0-SNAPSHOT) The qualifier **SNAPSHOT** remains, until patch code line gets integrated to cons code line.

Integrating code lines and incrementing TwoGo versions

Code line integration

Integrating code lines is done via [SAP Perforce Management System](#).