

# What's new in 5.5 (October 2013)

IKAN ALM offers a secure and flexible process-centric Application Lifecycle Management solution for both local and distributed development teams, and manages and automates SOA, Agile and traditional development processes. It complements existing version management tools by automating the complete software lifecycle management process, offering a single point of control and delivering support for your build, deploy, release and software lifecycle management and the associated approval processes. IKAN ALM helps reducing build complexity, solving complex deployment issues and accelerating release cycles.

A tailored lifecycle from development to production, including test and quality assurance, can be implemented, offering a comprehensive framework across all major platforms including Windows, UNIX, Linux and even z/OS mainframe systems.

**IKAN ALM 5.5 includes the following major enhancements.**

## Custom Phases

Custom Phases are the major innovation for the IKAN ALM 5.5 release. By creating Custom Phases, users can extensively customize the workflow of their projects, using highly reusable building blocks. Users can create a Phase, upload their scripts (ANT, NANT,) into it, define the Phase Parameters their scripts react to, and then use that Phase in their Projects. By using the import/export features, Phases can not only be shared between different Projects, but also between different IKAN ALM installations.

IKAN ALM comes bundled with a number of "Core" Phases which perform the basic functionality of Application Lifecycle Management, such as retrieving and tagging code from a versioning system, managing build artifacts, Issue Tracking, etc... Furthermore, IKAN ALM provides a set of "Certified" Phases which have been developed and tested by IKAN. Those Phases are extremely useful to ease and automate the execution of important recurring tasks, such as deployments to Tomcat or IBM WebSphere, running HP ALM tests, z/OS integration, etc...

On top of that, users can create their own Phases which run their own scripts. The benefits of creating a Phase to perform a task instead of running one large monolithic script are:

- **Reusability:** Phases can be shared between projects and also between IKAN ALM installations.
- **Improved logging:** by using many small Phases, the workflow is split up into smaller fractions. That facilitates the understanding and monitoring of the workflow as it will be much easier to exactly detect which step in the workflow failed.

- **Versioning:** a Phase is identified by a unique name/version combination. When the scripts inside a Phase change, the version number will change as well. This enables users to see exactly which version of a script they are running, and also enables them to use different versions of the same Phase in one single IKAN ALM installation.
- **Maintainability:** by using many small, limited-purpose Phases, the scripts being used tend to be smaller, what makes them simpler and easier to maintain.
- **Improved management of parameters:** by formally declaring the parameters the scripts of a Phase react to, it is easier to set the required parameters. On top of that, IKAN ALM contains features to do mass manipulation of parameters.

### **Modular (OSGi) architecture Server and Agent daemon**

The architecture of the Agent and Server daemon has been completely refactored in order to support Custom Phases. The daemons now run in an OSGi container (Apache Karaf) whose Service Registry controls the dynamics involved in such Phases: missing Phases will be automatically distributed from the central Phase Catalog and activated in the daemons, on local and remote systems. It is possible to patch or roll out new Phase versions, without impacting the running processes. Finally, the OSGi container allows running different versions of the same Phase on one Server or Agent.

### **Ability to specify the log format of Scripting Tools**

It is now possible to specify the format of the log generated by a Scripting Tool (ANT, NANT, Maven2,). The format can be set to TXT or XML. TXT logs have the benefit that they are smaller and that they can be read while the Tool is being executed. The view of XML logs can be customized in the IKAN ALM GUI by applying different XSL templates.

### **Enhanced functionalities**

- Support for Java 7 and OpenJDK 6 and 7
- Support for Windows 8 and Windows Server 2012
- Native 64-bit Windows installers

### **Bug fixes**

Several small bug fixes and optimizations have also been implemented. Refer to the readme of the installed product for more details.

Get the latest version on: <http://www.ikanalm.com>