

# **IKAN ALM Technical Sheet**

Release 5.8

January 2018



# **IKANALM**

IKAN Development N.V.  
Schaliënhoevedreef 20A  
2800 Mechelen  
BELGIUM

© 2006 - 2018 IKAN Development N.V.

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, for any purpose, without the express written permission of IKAN Development N.V.

The IKAN Development and IKAN ALM logos and names and all other IKAN product or service names are trademarks of IKAN Development N.V.  
All other trademarks are property of their respective owners.

---

# Table of Contents

<b>Chapter 1 - Overview .....</b>	<b>1</b>
<b>Chapter 2 - Technical Features .....</b>	<b>2</b>
2.1. Build Features.....	2
2.2. Deploy Features .....	2
2.3. Support for Lifecycle Management .....	3
2.4. User Interface .....	3
2.5. Support for Version Control Systems .....	3
2.6. Version Control System Interaction .....	4
2.7. Ant/Gradle/NAnt/Maven 2 Builder/Deployer .....	4
2.8. Supported Transfer Protocols .....	4
2.9. Issue Tracking .....	4
2.10. Package-based Build and Deployment .....	5
2.11. Build/Deploy Notification .....	5
2.12. Audit Trail .....	5
2.13. Modular Architecture .....	6
2.14. Installation.....	6
2.15. Security - Integration with JAAS .....	6
2.16. Commandline Interface .....	6
2.17. Customizable Reporting.....	6
<b>Chapter 3 - Technical Requirements.....</b>	<b>7</b>
3.1. Minimum system requirements .....	7
<i>Required Java Environment.....</i>	<i>7</i>
<i>Supported Application Servers .....</i>	<i>7</i>
<i>Supported Production Databases .....</i>	<i>7</i>
<i>Supported Operating Systems.....</i>	<i>7</i>
<i>Client Web Browser (Java Script must be enabled).....</i>	<i>7</i>
<i>Supported Version Control Systems .....</i>	<i>8</i>
<i>Supported Build Tool Integrations.....</i>	<i>8</i>
<i>Supported Deploy Tool Integrations.....</i>	<i>8</i>
<i>Supported Mail Integrations.....</i>	<i>8</i>
3.2. Minimum Hardware Requirements.....	8
<i>IKAN ALM Server.....</i>	<i>8</i>
<i>IKAN ALM Agent.....</i>	<i>8</i>

# CHAPTER 1

---

# Overview

IKAN ALM is the most technologically advanced solution on the market today for managing software change, offering a uniquely integrated web-based Application Lifecycle Management (ALM) platform. IKAN ALM combines both continuous integration and lifecycle management, offering a single point of control and delivering support for your build, deploy, release and software lifecycle management processes. By using existing technologies to extend the development environment, IKAN ALM continues where these best of breed products stop.

With IKAN ALM:

- developers can be kept up-to-date of (un)successful builds,
- team managers are kept up-to-date of the delivered builds into the live environments,
- configuration managers can oversee the status of the environment and know what will change or was changed at a point of time,
- project members can approve/reject scheduled changes or be notified when changes are going to take place.

IKAN ALM offers an open framework for realizing your IT Best Practices implementation for Application Lifecycle Management in conjunction with existing versioning, build and deployment tools without heavy and costly migration efforts.

Do not hesitate to contact us to learn more about IKAN's solution for Application Lifecycle Management.

---

# Technical Features

## 2.1. Build Features

- Support for multiple projects
- Continuous Integration: temporal scheduling of project builds
- Ability to manually initiate a build
- Build/rebuild any project version
- Build/rebuild on multiple environments
- Customizable the build workflow: adapt the flow of sub-processes (Phases) to be executed for a build
- Dynamic and secured build parameters which may be edited just before the execution of the build
- Configurable VCR tagging: template-based tagging, defined for each Project Stream (with variables such as project name, build number, date/time, branch prefix, suffix, ...). The default (template-based) tagging may also be overwritten at build creation, e.g., to enable applying labels like “RC1” or “BaseLine4.0”
- Project dependency:
  - include the latest result, a specific result or the production build result of a dependent project
  - include and build the latest source or a specific production source version of a dependent project
- Incremental build:
  - only build the modified sources compared to the previous build
  - only build the modified sources compared to the build in production
  - only build the modified sources compared to a specific VCR tag
- Tag-based build: Build based on a specific tag instead of on the latest code

## 2.2. Deploy Features

- Deployment of successful builds
- Temporal scheduling of project deployment
- Ability to manually initiate a deployment
- Deployment of any Project Version
- Deployment on multiple environments, in sequence or in parallel
- Customizable Deploy workflow: adapt the flow of sub-processes (Phases) to be executed for a Deploy
- Dynamic and secured deploy parameters, may be edited just before the deploy
- Partial deploy: deploy only the modified result files comparing to the previous build result

## 2.3. Support for Lifecycle Management

- Rebuild of previous builds on other target platform
- Deployment of previously executed builds to other target servers
- Define process flow: how builds migrate from BUILD to TEST to PRODUCTION environments (called Levels in IKAN ALM language)
- Customize the Build and Deploy steps for delivering to a Level
- Possibility to define hierarchical approval process after and/or before deployment towards a Level
- Possibility to define required notification after and/or before build/deployment towards a Level
- Deliver/Redeliver/Rollback functionality of built/deployed software
- Freeze functionality blocks the continuous build process or any other new build on the build level
- Optional Levels: possibility to skip a Level in a Lifecycle
- “Milestone” Levels: levels with no associated environments to support an advanced Approval and Notification process
- Define one or more physical environments (Windows, Linux, Mainframe, ...) for each level

## 2.4. User Interface

- Web-based user interface allows configuration of the entire system (no need to manually edit configuration files)
- History log of all administrative changes ensuring maximum traceability
- Convenient list of all the projects and color coded results of their last builds (succeeded/failed)
- Overview pages with enhanced search capabilities to quickly find the status and history of a Project's lifecycle.
- Customizable desktop providing a clear status view of the user's projects
- Two-way overview of Build dependencies.
- Convenient interface for approving, rejecting and searching approvals.
- Intranet project with javadocs, JUnit test results, metrics, and downloadable artifacts
- Interface to quickly identify and delete obsolete build artifacts
- Internationalization: the interface language can be defined through the user definition
- Context-sensitive help is available in HTML format
- Convenient interface to easily re-use defined build or deploy parameters through cloning of parameters and environments, or by defining machine parameters that are inherited by the environments

## 2.5. Support for Version Control Systems

- Subversion
- Git
- Microsoft Team Foundation Version Control (TFVC)
- CVS
- IBM Rational ClearCase (UCM and Base)

- Microsoft Visual SourceSafe
- Serena PVCS Version Manager
- Support for any Version Control System can be made available via the API interface

## 2.6. Version Control System Interaction

- Performs checkout or sync of project sources
- Applies a label with the current project version number
- Diff with the previous built/deployed software
- Show Sources (with version info) related to a build or deploy
- Ability to select and move individual files or components through a lifecycle (Subversion, TFVC and ClearCase)
- Subversion meta properties are automatically retrieved during the "Retrieve code" phase and can be used by the build or deploy scripting tool

## 2.7. Ant/Gradle/NAnt/Maven 2 Builder/Deployer

- Ability to execute any Ant/Gradle/NAnt/Maven 2 script to build/deploy the project
- No intrusion into the build/deploy script
- Ability to pass extra Ant/Gradle/NAnt/Maven 2 properties and change them right before build/deploy
- Wrap an Ant/Gradle/NAnt/Maven 2 script in a Custom Phase with specific properties and easily distribute and reuse it in different Projects or IKAN ALM installations
- Ability to specify JVM options like -Xms and -Xmx
- Support for Multiple Versions of Ant/Gradle/NAnt/Maven 2
- Configurable Ant/Maven 2 Classpath
- Ability to run Ant via the "Ant Launcher"
- Support for Multiple JDKs or .NET frameworks

## 2.8. Supported Transfer Protocols

- FileCopy using Windows shares and Linux mounts
- Secured file transfer using Secure Copy (SSH)
- FTP

## 2.9. Issue Tracking

- Link builds with issues, manually or automatically
- Possibility to link issues with builds manually
- Synchronize issue properties (status, description, etc.) with external Issue Tracking System

- Automatically add comments to issues each time a build moves through the lifecycle
- View details of issues from within the IKAN ALM GUI
- Built-in integration with JIRA, Team Foundation Server, HP Quality Center and CollabNet TeamForge
- Integration with any 3rd party Issue Tracking System via API

## 2.10. Package-based Build and Deployment

- Ability to select and move individual files or components through a lifecycle (Subversion, TFVC and ClearCase). Supports mainframe way of development
- Specific revisions/versions of individual files can be chosen
- A history log is maintained, allowing auditing the changes in the content of a Package (files added/removed/modified)
- A Package Build Group concept handles dependencies between Packages

## 2.11. Build/Deploy Notification

- Support for:
  - SMTP
  - NetSend
  - RSS
- Customizable, internationalized templates for plain text or HTML format e-mails
- Send out e-mails informing interested parties of build/deploy results
- Receive e-mails for every build/deploy, only failed build/deploy or only successful build/deploy
- Notification when build/deployment is requested/executed
- Notification when approval is requested/executed
- E-mails contain hyperlink which automatically forward to detailed information in the user interface
- RSS Feeds for displaying data about the last 10 builds/deploys that meet specified criteria

## 2.12. Audit Trail

- All administrative changes get a version log
- Build/Deploy logs for every build/deploy of every project
- Status overview of which build is residing at which machine/level
- History of built/deployed software through the lifecycle of a project
- Source-to-load relationship: overview of which sources have been used to make the build result, and which binaries have been deployed

## 2.13. Modular Architecture

- Agent/Server Architecture enabling distributed builds and/or deploys
- Possibility to define multiple build servers
- Possibility to define multiple deployment agents
- Command Line Interface enabling creation of builds and/or deploys on external triggers

## 2.14. Installation

- Graphical or text-based installer for the full product

## 2.15. Security - Integration with JAAS

- Integrates with operational security system
- User Authentication
- Passwords and passphrases in all IKAN ALM settings (System Settings, VCR definition, Secure Shell...) cannot be read by any IKAN ALM user
- All secure values are encrypted before they are stored in the database for maximum security
- Possibility to encrypt all communication between IKAN ALM components
- User Authorization:
  - For requesting builds
  - For requesting deliveries to levels
  - For approving/rejecting builds and/or deploys
  - For administering projects
  - For administering IKAN ALM

## 2.16. Commandline Interface

- Execute a build or deploy from a command prompt
- Allows scheduling builds and deploys with an external scheduling tool (Task Scheduler, Cron, ...)
- Generate a report from a command prompt
- Create/edit a package from a command prompt

## 2.17. Customizable Reporting

- The build and deploy overview is available as a customizable report in several formats (pdf, csv, xls, rtf). These reports can be launched from the User Interface or Command Line Interface. Command Line reporting enables the integration with other tools (e.g. planning) or the creation of reports in batch.
- The ALM Reports tool allows generating several other administration or follow-up reports using a wide range of selection criteria.



# Technical Requirements

## 3.1. Minimum system requirements

### Required Java Environment

- A Java Server Runtime (JRE) or SDK version 1.7 or 1.8.
- The Oracle JVM and the OpenJDK JVM are supported.

### Supported Application Servers

- Tomcat 7.0 or higher, recommended 7.0.82, 8.5.23 or higher

### Supported Production Databases

- MySQL
- Oracle Database
- Microsoft SQL Server
- IBM DB2 for Linux, Unix and Windows

### Supported Operating Systems

- Windows 7/2008 R2 Server
- Windows 8/2012 R2 Server
- Windows 10/2016 Server
- Linux: tested on Redhat Fedora, CentOS, Suse Linux Enterprise, OpenSUSE, Debian, Ubuntu, Mandriva, zLinux. Other recent Linux distributions should also work.
- Unix flavors: HP Unix, Solaris, IBM AIX etc.

### Client Web Browser (Java Script must be enabled)

- Internet Explorer 10.x, 11.x, Microsoft Edge
- Mozilla Firefox
- Google Chrome

### Supported Version Control Systems

- Subversion 1.4.x, recommended 1.6 and higher

- Git 1.8 and higher
- Microsoft Team Foundation Version Control (TFVC) 2013
- CVS 1.x, recommended 1.12 and higher
- CVSNT 2.x, recommended 2.5.x and higher
- IBM Rational ClearCase (Base and UCM) 7.1 and higher
- Microsoft VisualSource Safe 6.x, 2005
- Serena PVCS Version Manager

### Supported Build Tool Integrations

- Ant 1.4.x, recommended 1.9.2 and higher
- Gradle 2.2 and higher
- NAnt 0.85, recommended 0.90 and higher
- Maven 2.0.x, 2.1.x

### Supported Deploy Tool Integrations

- Ant 1.4.x, recommended 1.9.2 and higher
- Gradle 2.2 and higher
- NAnt 0.85, recommended 0.90 and higher
- Maven 2.0.x, 2.1.x

### Supported Mail Integrations

- SMTP

## 3.2. Minimum Hardware Requirements

### IKAN ALM Server

- RAM: minimum 512 MB (megabytes), recommended 2 GB (gigabytes) or more
- Disk Storage Space: minimum 10 GB (gigabytes), recommended 40 GB (gigabytes) or more

There is no hard and fast rule for disk storage space. The actual amount you will require depends on the number and size of projects managed with IKAN ALM, and the size of the build results stored in the build archive. The more projects and build results, and the larger they are, the more disk storage you will require.

### IKAN ALM Agent

- RAM: minimum 512 MB (megabytes), recommended 2 GB (gigabytes) or more
- Disk Storage Space: minimum 1 GB (gigabytes)

Just as for the IKAN ALM server, these are just indicative values. For a more mathematical approach, we refer to the article “Capacity Planning For Software Build Management Servers” on the “CM Crossroads” site: <http://www.cmcrossroads.com/cm-journal-articles/6671-capacity-planning-for-software-build-management-servers>.