



Autobuild is an enterprise build and deployment automation solution for Java EE, WebSphere® Portal and Process Server applications.

Autobuild automates the entire application development lifecycle: compile, package, deploy and test. This includes automated deployment to IBM WebSphere® line of products and their automated configuration. Autobuild eliminates the need for custom build scripts and complex configuration files. After Autobuild is made aware of deployment environments, configuring a project's build process requires little to no effort.

AUTOBUILD CAPABILITIES

- **Build configuration without Ant scripts, XML or property files.**

Autobuild infers most of the information about projects and their environments from Eclipse project files, project's directory structure, application server parameters and other existing configuration artifacts. For example, Autobuild obtains information about compile and package dependencies from the classpath defined in Eclipse. Autobuild decides how to build a project based on the collected information and "hints" optionally provided by developers.

- **Flexible framework integrated with an open source build server.**

Out of the box, Autobuild is integrated with a customized version of an open-source build server, [LuntBuild](#). LuntBuild provides support for continuous integration, notifications and other build management tasks. Use of other build servers is also possible. All AutoBuild functions can be customized and extended using an Ant-based plug-in mechanism.

- **Support for build and packaging of JAR, WAR, EAR, portlets, portal themes, SCA applications, Web services, EJB 2/3.**

Autobuild performs all standard build tasks, including Java compilation, XML validation, JSP compilation, packaging in JAR, WAR and EAR formats. Additionally, Autobuild supports static code analysis via its integration with [FindBugs™](#) tool. If needed, Autobuild will run WebSphere® tasks to generate the code, for example serviceDeploy runs automatically as part of SCA application packaging before deploying the application to WebSphere® ESB and Process Server.

Autobuild supports packaging of portal custom themes and "wps.ear" file format.

- **Automated deployment to WebSphere® Application Server 6.0/6.1, Portal Server 6.0, Process Server 6.1**

Autobuild installs applications to target application servers or clusters, distributes applications to nodes and starts them up. An application can be installed on any combination of clusters and servers, including the support for multi-clustered WebSphere® Portal cells. Portlet applications, portal themes and individual portlets are automatically registered in WebSphere® Portal. Custom [XML configuration interface](#) (a.k.a. XMLAcces) files to create portal pages are supported as well.

- **Support for unit testing, functional and integration testing.**

Autobuild supports JUnit, HTTPUnit, [Canoo WebTest](#) and [Eviware soapUI](#).

Tests run automatically during the build (JUnit) and after the application is deployed (WebTest, soapUI). Test suite to run depends on the environment. Test reports are published on the build server.



- **Support for environment-based release management.**
Each environment consists of one or more WebSphere® Application Server cells. Many different release schemes can be supported, the most typical being building in development and promoting to test, acceptance and production environments.
- **Support for project dependencies and group builds.**
Autobuild automatically re-builds dependent projects and updates EAR or WAR files with the artifacts from the dependent builds.
- **IT operations-related features.**
Autobuild provides monitoring scripts that can perform "health check" of application servers. Autobuild automatically creates application server resources, such as data sources and JMS destinations. Autobuild can also perform periodic backups of WAS, Portal and DataPower configurations.
- **Support for WebSphere® DataPower XML appliances.**
Autobuild provides targets for uploading WSDL, schema and XSLT files to an appliance as well as targets for managing appliance configuration, including propagating a configuration baseline between environments (different devices).

AUTOBUILD BENEFITS

- **Consistent and fully automated build and release processes improve software quality.**
- **Autobuild eliminates the need for investments and resources required for developing and maintaining an in-house build system.**
Developing an in-house custom build system is a significant undertaking that requires rare-to-find combination of skills, including Java EE, WebSphere® Application Server and Portal scripting, Configuration Management and Ant.
- **Improved collaboration**
All artifacts and reports are available from the build server, as illustrated by the screenshot below:

The screenshot displays the Autobuild web interface. At the top, there's a header with the 'Autobuild' logo and version '1.5.3'. Below the header is a navigation menu with tabs for 'Builds', 'Projects', 'Users', 'Properties', and 'Administration'. The main content area shows details for a build labeled 'helloWorldWeb-1.0-20080405-215717'. The build status is 'success'. A table lists build artifacts:

Filename	Size	Last modified
helloworld-1.0-src.zip	81335 bytes	2008-04-05 21:58
helloworld.war	39708 bytes	2008-04-05 21:58



- **Easier agile development adoption thanks to Autobuild support for key agile practices.**
- **Faster turnaround for application changes as the time required for application releases is greatly reduced.**
- **Elimination of manual deployment tasks reduces error rate and improves SLAs.**

A complex application may consist of many different units of deployment (EAR files, WAR files) each having its own deployment parameters (e.g., target servers or clusters). Deploying such an application manually is error prone and time consuming.

Autobuild is capable of automating deployments of applications of any complexity.

- **Web-based build server interface greatly simplifies build management.**

No need to deal with hard-to-remember command line parameters and environment variables, all builds are configured using Web-based UI similar to the one below:

Continuous to Dev	
Execution status	success at 2008-04-09 23:17
Description	Label format
Next build version	helloWorldWeb-\${project.var["version"]}-\${system(year+numericMonth+dayOfMonth+"-"+hour+minute+second)}
Variables	env.name=dev
Work subdirectory	
Trigger type	simple
Repeat interval(minutes)	15
Build necessary condition	vcsModified ——— Continuous builds are triggered by VCS changes
Associated builders	BDF Ant Builder
Associated post-builders	
Build type	clean
Post-build strategy	do not post-build
Label strategy	label if success ——— Successful builds are labeled automatically
Notify strategy	notify when failed or status changed
Schedules current schedule depends on	Builds can be chained together if needed
Dependency triggering strategy	trigger schedules this schedule depends on
Build cleanup strategy	do not cleanup builds automatically
Latest build	helloWorldWeb-1.1-20080407-215016 success at 2008-04-07 21:51

- **Improved security and accountability,**

Access to builds and deployments is controlled by the build server security which can be configured at a project and environment level. For example, deployments to production can be restricted to authorized data center personnel.

Build server's audit capabilities can be used to find out who executed a specific build or a deployment.

AUTOBUILD



Autobuild is available as part of a "turnkey" solution provided by MyArch, Inc., developer of Autobuild. MyArch specialists will analyze your applications, configuration management processes and operational environment and install and configure Autobuild to meet your needs. At the end of the engagement, you will be able to build and deploy your Java and Java EE projects in a **fully automated way**, including running continuous builds, creating releases and promoting the releases to "higher" environments, such as user acceptance or production.

A typical Autobuild engagement takes only a few weeks; the duration depends on the complexity of your environment. The Autobuild software components (the build framework and the build server) are open source and free; you will be paying only for Autobuild configuration and customization necessary to meet your requirements. For example, Autobuild configuration includes defining parameters of your environments, such as URLs of WebSphere® Deployment Manager.

Additionally, MyArch provides comprehensive support services for Autobuild implementations tailored to your specific needs.

If you have questions about Autobuild or would like to schedule an Autobuild demonstration, please send us an email at autobuild@myarch.com or give us a call at 703-608-9302.