

AceProposal

AceProposal

Abstract

Apache Ace is a software distribution framework based on OSGi that allows you to manage and distribute artifacts, like e.g. software components.

Proposal

Apache Ace is a software distribution framework that allows you to centrally manage and distribute software components, configuration data and other artifacts to target systems. It is built using OSGi and can be deployed in different topologies. The target systems are usually also OSGi based, but don't have to be.

Background

When assembling software out of reusable components, the task of deploying software onto an ever increasing number of targets is not trivial to solve. This becomes even harder when these targets require different components based on who's using them.

A key technology in the Java space for developing component based applications is OSGi. The OSGi specification, which has been around since 1999 and by now has matured into the de facto module system for Java, allows you to write components that can interact through services and that allows components to be updated individually, without disturbing the rest of the components.

Although the OSGi specification describes how software distribution should be done, it does not actually prescribe any protocols or implementations. Apache Ace implements a software distribution system based on, but not limited to OSGi. It is setup so it can deal with different target types, using different protocols. Also, it can handle an extensible number of artifact types (bundles, configurations, resources, ...).

Rationale

When you start using OSGi to build reusable components, the task of managing those components and their use in various applications becomes non-trivial. Apache Ace allows you to group those components and assign them to a managed set of targets. This allows you to distribute updates and new components easily, while keeping a full history of what was installed where during what period. It also helps you setup an automated development, QA /testing, staging and production environment.

Initial Goals

The initial goals for Apache Ace are:

- Donate the existing codebase and import it.
- Setup the incubation infrastructure (svn repository, build system, website) so we can run continuous builds with automated tests and publish all available documentation.
- Get people involved in advancing the code base in different directions, integrating it with other projects at Apache.
- Prepare for an initial release that demonstrates the systems core capabilities.
- Present the project to the community at [ApacheCon](#) 2009 US.

Current Status

The current codebase is developed and tested in various configurations. It was developed at luminis over the last couple of years using Scrum, so we have internally demonstrated we can release often and produce working code using a transparent process. Documentation for the project is now available on an internal wiki, which can be donated and converted to the Apache Ace website. We did not yet use mailing lists as the primary collaborative process, as the whole team met face to face on a daily basis.

Meritocracy

Some of the core developers are already committers and PMC members at Apache, so they understand what it means to have a process based on meritocracy.

Community

In the past, luminis has been talking to various interested users and developers about Apache Ace, and we believe there is an interest in this project. Feedback at [ApacheCon](#) EU 2009 and afterwards on the Apache Felix mailing list confirmed that. The problem that is being solved is one that many software developers run into, so it should appeal to them. Our list of initial committers already includes people from different backgrounds.

Core Developers

The core development team is a mix of people that work for luminis and have been involved in the project up til now and new committers, some of which have previous experience at Apache.

Alignment

The initial codebase makes use of Apache Felix as its core framework. It also uses various other components of that project. As a project that builds on components we are constantly looking out for existing components that can accelerate our implementation and we want to actively work with other projects to make that happen. For building and testing we use Apache Ant and have developed a couple of extensions that help managing OSGi projects. Finally, at least informally a couple of projects at Apache have already expressed interest in a system that can help them do software provisioning.

Known Risks

Apache Ace uses Felix as its default OSGi implementation and some of its developers are also part of the Felix community. We are open to collaborate with other Apache projects, looking at candidates such as Commons, Sling, [JackRabbit](#) that could help us in certain parts of our infrastructure.

An important reason for open sourcing this project at Apache is the strong community, as well as the Apache license. This will attract more users and developers so the project can be moved forward into new directions that we would otherwise not have been possible. Judging from the initial interest from some of the other projects at Apache, we certainly see ways in which to collaborate and advance the project, possibly in ways we would never have thought of. However, we have been able to support and develop this product outside of Apache quite well, so in no way are we trying to just dump the code there or merely trying to generate publicity.

Initial Source

Apache Ace has been in development within luminis since 2005.

Source and Intellectual Property Submission Plan

The current codebase is owned by luminis, and will be donated together with its documentation. We will get the paperwork out of the way as soon as possible. There should already be a CCLA on file for luminis and the people that are already involved with Apache obviously have ICLAs on file.

External Dependencies

There are quite a few open source libraries already used. The libraries, their sources and licenses are listed here:

Apache Felix, ASL:

- framework
- shell
- shell-tui
- obr
- http.jetty
- config admin
- event admin
- deployment admin
- dependency manager
- prefs
- upnp.basedriver
- org.osgi.foundation
- core
- compendium
- javax.servlet

Apache Ant, ASL:

- ant.jar
- ant-contrib.jar

Apache Velocity, ASL:

- velocity

KXML2 (<http://kxml.sourceforge.net/kxml2/>), BSD license:

- kxml2 (hmm, what was that issue we had with that in felix)

Knopflerfish (<http://knopflerfish.org/>), BSD style license:

- log_all.jar, useradmin_all.jar

Luminis Open Source Server (<https://opensource.luminis.net/>), BSD license:

- net.luminis.build.plugin.jar

XStream (<http://xstream.codehaus.org/>), BSD license:

- various xstream jars

Required Resources

Mailing lists:

- ace-private
- ace-dev
- ace-commits
- ace-user (only after leaving the incubator)

Subversion:

- <https://svn.apache.org/repos/asf/incubator/ace>

Issue Tracking:

- JIRA: Apache Ace (ACE)

Wiki:

- Confluence: Apache Ace (ACE)

Initial Committers

These existing Apache committers have either worked on the initial codebase (Christian, Karl and Marcel) or expressed an interest in extending the project:

- Marcel Offermans
- Karl Pauls
- Christian van Spaandonk
- Clement Escoffier
- Felix Meschberger
- Carsten Ziegeler

Community Members

The following people have already expressed their interest in actively participating in this project:

- Bram de Kruijff
- Toni Menzel
- Alin Dreghiciu
- Dennis Geurts

Affiliations

For the record, Marcel Offermans, Christian van Spaandonk and Dennis Geurts work at luminis and might get paid to do certain work on Apache Ace.

Sponsors

We have approached both the champion and an initial list of mentors that have agreed to mentor this project.

Champion:

- Karl Pauls

Mentors:

- Carsten Ziegeler
- Niclas Hedhman
- Bertrand Delacretaz

Sponsor:

- Apache Incubator