

Community Roadmap

Summary

This is what we think should be the Community Roadmap for 2012. Suggestions people made in details appears below.

- Finishing the tutorials
- Writing a Getting Started with Project Wonder book
- Improvements on the development and deployment tools to make them easier to work it (especially on starting up, like an installer than install Eclipse, WOLips and the WO/Wonder frameworks)
- Fixing bugs or doing improvements that are affecting at least 25% of the community (the 2G limit of WOFileUpload, model reverse engineering bugs, etc.)
- Documentation about how to move to Wonder APIs instead of the straight WO APIs (that's the first part before moving to alternatives)
- Better integration between WO and iOS/Android

In details

Development tools

[ERRest templates \(done!\)](#)

[ERRest and Veogen integration](#)

It would be nice to integrate Veogen with ERRest, e.g. when you call Veogen for the EO entities, it could create base REST controllers for the default actions at the same time.

[Command-line WOLips](#)

It would be cool to have a command-line version of WOLips that could do:

- Create a basic Eclipse project so that you can work with it by command-line or in Eclipse
- Edit the EOModel by command line
- Call Veogen
- Do a reverse engineering of a RDBMS
- Do component validation

An example of such a thing is the Play! Framework. A command-line version will allow people to create quick REST projects with just Ant and a couple of JARs.

[Enhance component validation for WOLips](#)

(even though it is not crashing, but refreshing a workspace I usually have 1 or 2 components where validation fails, and they are not the same components each time, weird).

[ERRest wizard for integration into existing project.](#)

I would like to get more details about that, since to integrate ERRest in a existing project requires only that you add the framework and register the request handler -- Pascal

[move rulemodeler into eclipse](#)

Or at least have a RuleModeler that can work on Windows or Linux -- Pascal

[fix EntityModeler support for reverse engineering](#)

We need more details -- Pascal

[give EntityModeler the options EOModeler had for rev. engineering existing tables](#)

you could have it auto-generate the class name from the table name instead of just extending EOGenericRecord like it does now

side-by-side view of .html/.wod file that can be in a different window than the .java file

Could be a preferences like it is for Entity Modeler -- Pascal

fix the bindings view (take it out of the experimental stage)

Fix the Bindings view and make it the default tab. It could be a very powerful view for beginners.

A DBEdit-like application

It would be nice to have a DBEdit type of application so that you can query the database by EOF instead of using tools like pg_admin. Maybe by using D2W? -- Pascal

WOBuilder type app

The classic 😊 I think having a better Preview tab and fixing the Bindings view would make it easier for everyone. -- Pascal

Something to eliminate the 'get' prefix in the auto accessor generation within Eclipse java files.

I think Mike said it was not possible to change that behavior. We will need to look at Eclipse dev stuff. -- Pascal

Support something other than Eclipse, like IntelliJ integration.

I guess the only thing missing is component edition/validation. Entity Modeler is already available as a standalone app. -- Pascal

Improve the support for JAR dependencies in WOrder and WOLips.

We need more details on that. -- Pascal

Better Maven support.

At the very least, it would be nice to be able to launch my projects in the debugger and not have to set the class path each time I hit a breakpoint to view the source. But it's a minor inconvenience.

Improve the support for Maven in WOLips. At least, no bugs.

A mechanism to help processing web resources. It is hard to use tools like LESS or JS minification with the current set of tools.

We need more details on that. -- Pascal

One click installer using Java Web Start

A one click Java Web Start installer. Just click on a hyperlinked .jnlp file and Java web start downloads the installer which then would install Eclipse, WOLips, WO 5.4.3, jadclipse, jad, latest svn, subclipse, Wonder 54 branch, WO javadocs, set up a new workspace complete with working sets for Wonder frameworks, examples, applications, and tests. 'Advanced' options could be offered, including apache/monitor/wotaskd installation and setup.

Must check installer from Ken Ishimoto. Expending Golipse to install the core frameworks and Wonder is also an option.

Simple D2W JavaApplet to configure a DirectToWeb app

A fixed Direct To Web JavaApplet should be reinstated as a simple way to configure a DirectToWeb app. Rule Modeler is OK for advanced rulemodeling, but is too complicated for beginners with DirectToWeb.

Perhaps inline forms would be easier than an applet. For example, right now, if you hit the D2W link on the debug panel, you see tables at the page level and on property names. Perhaps it would be easy enough to simply provide a form + table instead of just a table. Then you could select the component name from a popup menu or add/remove displayPropertyKeys from the page level form.

I am still yearning for the old drag and drop tools and a ruleeditor for D2W apps like the old java app that prevents me from shooting in the foot.

See the comments about RuleModeler -- Pascal

Build and Deployment

Build another Jenkins server to test patches

We should have a system that auto-patch Wonder on another server, so that we can see if the patches compiles fine and do basic testing to see if it's broken. If it works, we can commit the patch in Wonder.

Get javamonitor and wotaskd working on 64-bit windows (javamonitor has issues displaying web resources and wotaskd can't launch the 32-bit SpawnOfWotaskd.exe from a 64-bit jam)

Should the community buy a copy of Windows 2008 Server 64 bit? I don't have a copy, so I can't work on that. -- Pascal

Easier deployment

- Make it possible to create a double-clickable `.woa.jar/app` that would allow dead-simple "deployment" of an WebObjects Application, much like Jenkins can be downloaded and just started and it works.
 - Need more details. An `.woa` is already launchable by itself with the two scripts bundled in it. If someone wants to ship a WO app as a product, they can wrap it up in an installer or a package (`.rpm`, `.deb`, etc.) -- Pascal
- Deployment support built-in WOLips
 - It could be based on the `jsch` Ant task so that deployment is done by SSH + calls to `wotaskd` -- Pascal
- REST APIs directly in `wotaskd`
 - Having the REST APIs in `wotaskd` would also use to manage `wotaskd` with anything (JavaScript, `curl`, etc.) -- Pascal

Jenkins plugin to deployment from Jenkins to wotaskd

We need to take a look at other Jenkins deployment plugins for ideas. -- Pascal

RPM and deb packages to install wotaskd/Monitor/mod_http on Linux + central repository

It would be nice to have RPM and DEB packages for the deployment tools so that installation of the tools can be done in one step. We could build those packages with Jenkins. -- Pascal

OpenVZ (CentOS and Ubuntu) templates with the tools built-in

OpenVZ allow people to run "virtualized userspace" (similar to FreeBSD Jails or Solaris Containers), it's chroot on steroids. -- Pascal

VMWare template with the tools built-in

Maybe it could be useful to have a VMWare template with the deployment tools built in. It could be based on Ubuntu or CentOS and have a link to the DEB or Yum repository. We might have issues with Java licensing (can we distribute the JDK from Oracle as part of the package?). -- Pascal

"Small" deployment service on WOCCommunity.org

That service would be based on OpenVZ and allow running small (512 MB of RAM max) applications with a custom front-end for managing instances. Would be especially useful for people that only want to have a REST backend for mobile apps without having to deal with RDBMS, `wotaskd` and Linux management. -- Pascal

Nagios plugin

That plugin would contact to the DirectActions and REST services from Monitor and `wotaskd` so that you can monitor if the app is dead or not. Could also query things like sessions and other metrics coming from `WOStats`. -- Pascal

Wonder

jQuery Ajax framework

According to the surveys, 46% of the community is already using jQuery. Having a AjaxjQuery framework should a top priority.

Splitting the Ajax framework in two frameworks

The UI part of the Ajax framework should be moved to a "PrototypeJS" framework so that the Ajax framework only have the Java code for the request handler and everything else.

ERSolr

It [already exists](#), we just need to add it to "master".

Enhance AjaxDatePicker to include a time picker

Override all WO components and EOF calls in Wonder

We should override all WO components and EOF calls in Wonder, so that it could be easier to move to another implementation. For example, we could later decide that instead of using EOF, we would use Cayenne, but by using calls from Wonder, the APIs in Wonder would talk to Cayenne instead of EOF.

Take over development of the original WO frameworks, perhaps gradually re-implementing them making the full stack free (as in speech) and open.

Begin reimplementing some key Foundation classes in WOnder that could use modernization and extension like NSArray.

Overhauling abandoned components and classes, which are in a semi-usable state.

I would like Project Wonder and WOLips to be not reliant on WebObjects. I think a significant amount of WebObjects functionality has already been overridden by Wonder. It would take years to exactly replicate WebObjects and that would include some undesirable "features". I like the idea of connecting Project Wonder and WOLips to something like a cayenne back end instead, even if it means I have to make some changes to my apps.

concurrency throughout EOF, including multiple DB connections per EOF stack

Can Kieran's session at WOWODC 2010 could be useful for that? -- Pascal

Adding a REST client in ERRest

We need a REST client in ERRest so that we can easily call REST APIs from other sources, and convert the results to EOs if required. A sort of client exists in the RestRoute example, and other code based on Jakarta httpclient was posted on the mailing list. We could take a look at Jersey and JBoss' RestEasy too.

Ted Archibald have sent a client based on httpclient in the mailing list on early February 2012.

Unit/Selenium tests for the core frameworks

We need unit/Selenium tests in Wonder for core stuff. Examples : doing fetchs with dates, testing prototypes on MySQL, PostgreSQL, H2 and FrontBase, core Ajax fonctionnality, etc. Those tests would be useful to test patches

Better integration with Cocoa and CocoaTouch

Since many WO developers also develop iOS or Mac apps, it would be useful to have better integration between WO and Cocoa/CocoaTouch. One idea would be to generate a Xcode project stub based of an EOModel and REST route, the Xcode project stub would have the necessary code to make REST calls to the WO app.

Excellent support for iOS REST backends

an example of integrating WebObjects with iOS, how to communicate from the iPhone app to the WebObjects backend.

I'd like to see a way to keep CoreData models and EO models in sync. Also a D2Rest from CoreData would be pretty cool.

Review all those TODO and FIXME comments in Wonder

Wonder source has accumulated many TODO and FIXME comments that should be reviewed if they

- a) are obsolete
- b) need code to be fixed/tested
- c) should have a JIRA created for to be fixed in the near future

Define a "WOCommunity formatter" for eclipse

Wonder code has been committed by many different people and all those people have different formatting preferences. By defining a general eclipse formatter the code would be cleaner and committing patches would (in the end) only show real code changes and no formatting changes.

serious release engineering

I say this year after year. The received wisdom of tracking the HEAD of the master branch is just not realistic for most businesses. Wonder needs proper release engineering.

Have a clear standard (stable version) for project Wonder (make sure the best option for each artefact is clearly identified).

Roadmap

Best D2W look frameworks possible!

fix EO model inheritance bugs

work on migration system

get db downgrades to work

better support for index creation and addition of constraints

more standardization of components

Apache Hadoop NoSQL support

Remove depreciated cruft in base frameworks.

more joda integration by default

Jabber InstantMessengerAdaptor

Cleaning up elements

Have all html elements be coded with enough CSS classes/ids so that the presentation/look can be fully controlled by CSS. One extreme anti-example is ERXCheckbox matrix which uses embedded tables... yuck! See <http://www.csszengarden.com/>

Localizing Wonder components that have static strings in it would be great. -- Pascal

Make a minimum version that just deals with the Apple bugs, and keeps WO up to date for people that don't use Wonder and use their own Frameworks instead.

wocommunity.org

Using a more widely seen video service for screencasts/podcasts

Evangelism: Use a more widely seen video service for screencasts/podcasts so that a broader group of developers can find them more easily. (A Vimeo channel for example)

Case studies on wocommunity.org

We need case studies on wocommunity.org. Something like <http://www.apple.com/business/profiles/> should be good, a one-page where people explain how WO is good for their business and how they use it.

Documentation

more docs and examples that are shorter and more focused

Better documentation that explains the used concepts in some of the frameworks.

the framework and tools should have more documentation about how I can use the different packages for. I know there are a lot of example projects, but you have yourself to dig into the code to see how things working and to understand how it is helpfull for my work.

better documentation, better clarification what is current, what is old stuff/deprecated. Sometimes the same topic is covered in several places - differently!!

D2W book

it would be a book about D2W i think that i have not been able to grasp "its greatness". i would like more documentation and examples (or another book) of how to make a successful framework that implements ajax functionality.

i like to have the information printed in books and for each of this books i would gladly pay 30-40 dollars.

i don't like how we depend on old books that mostly are directed to xcode, i own a copy a practical webobjects but a "starting in webobjects" or webobjects learning-style book is needed too.

Love the improvements to documentation that have been made on the wiki. It would be great to see this continue to improve-- especially with best practices for integrating/syncing with other systems (especially iOS) through REST services.

Better start up tutorials. Easy to understand Enterprise Objects because it is difficult to understand for new developers.

Documentation that better explains how it all fits together - there are a lot of really cool sounding frameworks, but its often not clear how to use them

(synchronization between app instances?)

Links from javadoc to wiki for more in depth conversation/examples.

More code examples. For example, the best learning tool for me was to go to the Ajax Examples, looking at the code samples and components. I learn more from there than just the bare-bone documentation on Ajax alone.

More examples

I would like to see finished applications including a web content management App and an eCommerce App available for download.

Performance issues/improvements: How to improve performance if there are any tips/ hints/ where we could put together a whole chapter there. (EOF, Ajax, etc).

More videos for beginners

Create official community documentation for essential steps in WebObjects installation / development / deployment

I think that one of the big drawbacks in the documentation right now is that there are too many references to historical configurations. It would be helpful to collate step by step documentation for the current stable versions of WebObjects, WOLips, MacOS X and release it as a community-sanctioned pdf.

For example at a minimum it would be helpful to have:

1. "Community Recommended Installation on MacOS X 10.6"
2. "Community Recommended Development Practices (running through Apache, bundle-less builds)"
3. "Community Recommended Deployment Practices"

New people don't need to know the history. They just need to know how they install WebObjects, Wonder, the tools, a database and get going for the current versions. This document set needs to be frozen and rereleased each time a major version change happens (whether Eclipse / WOLips, WebObjects, Wonder (affecting normal workflow) or MacOS X). That way we'll have documents archived for previous versions for those that need them, and straightforward current versions for everyone else coming to the platform.

Making Wonder independent from WebObjects is an excellent idea. Maybe it would enable a complete rewrite (one day).

Decide which is the best way to install WO and Eclipse/WOLips

We need to determine what is the best and "supported" way of installing WO and Eclipse/WOLips. Should we use WOInstaller.jar or the .pkg from Apple? Should we use Golipse for installing Eclipse and WOLips? When we figure out the best way, we have to update the wiki and wocommunity.org.

I 100% agree. There should be an official community supported method for installing eclipse / wolips / webobjects / wonder. All other flavours of installing stuff should be removed from the wiki.

As we also want to appeal to existing (Java-)programmers that already have their Eclipse environment set up it would probably make sense to have installation instructions / installers that only install the WO*bits into existing Eclipse.

New, modern tutorial

We need a long tutorial about Wonder and WebObjects. That tutorial should present how to do a "regular" WO app, a D2W app and REST services built on top of the same model.

I'd be happy to work on that one, provided some people can give me some advice on the stuff I do not understand, and comment. I started with a tutorial about two years ago but stalled because the second chapter should be about DirectToWeb, and I hoped that somehow I could work around RuleModeler until later in the tutorial, but a fixed D2W Applet never materialized. Just let me know if some people are willing to help.

Ken Ishimoto: We will record at the Bootcamp many Screen casts in Japanese for open releasing.

Assemble information regarding scalability of WO apps

Assemble information regarding scalability of WO apps i.e. what software/hardware measures can be used with pros/cons, scalability commandments (like the EOF commandments), code examples to help newbies for concurrency issues, ...

Write a WO/Wonder cookbook

I would like to see (and contribute to) a "cookbook" for WO programming tasks. There are existing examples (Ajax Example App) and I think cookbook organization (like the O'Reilly Cookbooks) would help those to be found and highlight needed recipes.

This is something I have been missing myself. I have put such cookbook code snippets into a text file that I regularly consult (yes, even after 9 years). Things like "How do I sort things" or "How do I filter objects", "How do I upload a file" are 1/2/3/4-liners in my reference. These things are trivial once you know them but for the beginner...?

I would gladly like to help here and hand over my snippets.

Linking the wiki and the Javadoc together

We need a way to link the JavaDoc and the wiki together. It could be special tags in the JavaDoc so that those tags are found, we automatically create a page in the wiki with a link to the JavaDoc.

Localized Javadoc

Would be nice to have a way for Localization Javadoc in Wonder. To Help people make Documentation in other Language too. For my Part about 40% Documentation in japanese is done, but no way to write or share.

Perhaps this could be achieved with html. `<p xml:lang="jp">Japanese Docs</p><p xml:lang="en">English Docs</p>` Then it's just a stylesheet switch when the language changes.

Japanese stylesheet like
*`[xml:lang="en"]`

Unknown macro: `{ display}`

For the outputting Javadoc that is a Solution, but I think there are 2 ways to lookup Javadoc.

1. Javadoc itself
2. In Eclipse hover over a Command

Other

more global training camps(I'm in mexico and i havent graduated yet from college so taking a flight to canada is not exactly a possibility here)

More simplicity

better way to pass on information changes and updates, easier to learn tools, easier way for individuals to help contribute.

To see more people regularly committing code, writing documentation and providing help. This seems to be happening recently.