

WOCCommunity 2014 Organization Survey

85 respondents.

Top 5 countries

Country name	%	# of developers
USA	32%	57
Germany	26%	56
Canada	12%	28
Australia	7%	15
UK	7%	26

For how many years has your organization been using WebObjects?

Less than 2 years	2%
Between 2 and 5 years	12%
Between 6 and 10 years	28%
More than 10 years	58%

Which versions of WebObjects does your organization use?

	2014	2013	2012	2011	2010
5.4.x	94%	94%	93%	87%	86%
5.3.x	12%	17%	24%	21%	35%
5.2.x	5%	10%	5%	3%	9%
4.5.x	1%	4%	1%	6%	6%
4.0.1	1%	1%	1%	1%	1%

Which platforms are used at your organization/department to DEVELOP with WebObjects?

Mac OS X	98%
Windows	8%
Linux	7%

Which platforms are used at your organization/department to DEPLOY WebObjects applications?

	2014	2013	2012	2011	2010
Mac OS X Server	60%	51%	52%	59%	68%
RedHat/Fedora/CentOS Linux	45%	45%	37%	40%	42%
Debian/Ubuntu Linux	27%	30%	23%	21%	10%
Windows	11%	15%	13%	14%	13%
Amazon Linux	12%	14%	12%	12%	2%
Mac OS X "client"	18%	10%	13%	9%	19%
Solaris	2%	9%	11%	7%	9%
FreeBSD	2%	6%	3%	6%	8%
SuSE Linux/OpenSUSE	1%	1%	1%	1%	0%

Which database systems does your organization use with WebObjects?

	2014	2013	2012	2011	2010
MySQL	52%	52%	45%	42%	39%

PostgreSQL	41%	51%	38%	44%	39%
Oracle Database	26%	24%	28%	28%	27%
FrontBase	16%	20%	19%	20%	22%
MS SQL Server	13%	9%	14%	14%	12%
H2	5%	5%	2%	6%	1%
OpenBase	6%	2%	6%	8%	8%
Sybase	1%	2%	1%	2%	2%
Derby	2%	2%	2%	1%	1%
Amazon RDS	0%	0%	0%	1%	0%
DB2	0%	0%	0%	1%	2%
HSQLDB	1%	0%	1%	1%	1%
SAP MaxDB	1%	1%	1%	1%	1%

Do you use Project Wonder in your project?

	2014	2013
Yes, in all of them	73%	66%
Yes, in some of them	19%	29%
No, but we will use them later	5%	0%
No, and we are not going to use it	1%	6%

Does your organization use DirectToWeb?

	2014	2013	2012	2011	2010
Yes	31%	39%	42%	42%	42%
No	62%	56%	45%	47%	58%
No, but planning in the near future	7%	5%	12%	12%	N/A

Does your organization consume SOAP services using WebObjects?

	2014	2013	2012	2011
Yes	28%	33%	32%	40%
No	68%	66%	59%	55%
No, but planning in the near future	4%	1%	9%	6%

Does your organization consume REST services using WebObjects?

	2014	2013	2012	2011
Yes	49%	43%	34%	28%
No	39%	39%	43%	50%
No, but planning in the near future	12%	18%	23%	22%

Does your organization provide REST services using WebObjects?

	2014	2013	2012	2011
Yes	55%	53%	46%	32%
No	32%	27%	32%	45%
No, but planning in the near future	13%	20%	22%	23%

Does your organization provide SOAP services using WebObjects?

	2014	2013	2012	2011
Yes	26%	25%	19%	24%
No	69%	73%	72%	69%
No, but planning in the near future	5%	2%	9%	7%

Does your organization use RIA technologies? If yes, which one(s)

	2014	2013	2012	2011	2010
AJAX framework from Wonder	73%	80%	71%	79%	71%
jQuery	67%	62%	45%	36%	N/A
Cappuccino	1%	1%	2%	6%	1%
Adobe Flex	6%	6%	5%	6%	6%
SproutCore	1%	2%	3%	5%	N/A
Dojo	4%	0%	2%	3%	N/A
GWT	4%	5%	4%	1%	2%
YUI		1%	1%	1%	N/A
Sencha Touch	1%	1%	1%	1%	N/A
MooTools	1%	2%	3%	N/A	N/A
DHTMLX	1%	1%	1%	1%	N/A
AngularJS	3%	2%	N/A	N/A	N/A
Backbone	3%	3%	N/A	N/A	N/A
MontageJS	1%				
Bootstrap	3%				

Do you use WebObjects as a back-end to other technologies, for example to communicate with a iPhone application? If yes, which technologies?

	2014	2013	2012	2011
iOS apps	81%	60%	32%	29%
Mac desktop (Cocoa)	21%	9%	10%	10%
Android	31%	4%	N/A	N/A
Windows Mobile	2%			
Windows desktop	7%			
JavaScript front-end	45%			

Does your organization still use WebObjects?

Yes, for new and current projects	73%
Yes, but only for maintenance of current projects	16%
Yes, but we are moving away from it in the next 12 months	6%
No, we moved away	2%
Not sure yet	1%

If you are stopping developing with WebObjects, what are the main reasons for dropping WebObjects at your organization?

I would use both WebObjects and Play

Lack of forward development or updates, bug fixes

Current WO apps work just fine, but we are experimenting with other frameworks for new ones. Key reason is that WO and Java are "not cool" and new hires prefer Rails, Django, etc. Another reason, is that there isn't much really interesting going on with WO development after we Mike left. That said, our main focus now is on client-side anyway (iOS, Android, JS).

three main reasons:

- 1 - we are moving all persistence to DynamoDB (negating need for EOF) and front ends to javascript clients (negating need for WO component frameworks)
- 2 - we are adopting continuous deployment with end-to-end test automation which is simply much easier to achieve with modern frameworks that are designed, built and maintained with such automation in mind
- 3 - good old availability of resource.

No Apple support

Shrinking development community

We are actively looking to replace WebObjects applications with more mainstream frameworks and open source solutions. We just don't feel like there is enough support and it is virtually impossible to hire replacements for our retiring WO coders.

Unable to properly scale stateful applications.

Lack of support from vendor.

Remaining community too small.

Employment candidates don't know WO.

Other frameworks have far surpassed WO in ease-of-use and functionality.

Closed source.

Architecture requires tight coupling to framework.

Too many bugs in both WO and Wonder; legal issues with reverse-engineering WO and fixing ourselves

Spending too much time fighting the framework to add new functionality.

Wonder project doesn't seem to consider the needs of large scale deployments

Coupling to ancient versions of frameworks (example: Axis 1.x)

Standards, best practices, and features that have already been developed by the Java community don't appear to be considered - understandable for old Apple code but baffling for new Wonder frameworks

Lack of Xcode, EOM, WebObjects Builder support in recent OSs.

Lack of Apple's support behind their technology

Not embracing current technologies.

1. Risks associated with losing WebObjects knowledge (developers retiring) and difficulties of finding WO developers or new hires who want to learn WO.
2. Desire of management to move to "standard" technologies (e.g. Rails, Java with Spring-Web) or newer frameworks like Node-js.

Uncertain future, and dwindling community. Brilliant technology, but core is not evolving because of Apple's abandonment. API compliant open source could change that. So could renewed Apple interest.

- Experienced staff unavailable on the job market
- Obsolete core, closed source
- Tempting competition (tapestry, spring. etc...)
- Small community in the brink of death (I know, we are not helping either... sorry)

We're not *planning* to jump ship (Where would we go?), but if we did, it would be because it's so difficult to find good developers who either (1) already know (or even have some basic familiarity with) WO, or (2) have any interest in learning it.

Reactive programming

Uncertain future. The situation would be different, when WO would be open source.

This was a primarily Windows shop and the other developers are unwilling to learn anything other than VB and maintain VB6 is still relevant and should be maintained. Having difficulty to even dragging them to at least C#.

We are still using WO for new development, but will seriously consider switching to something open source in the next 24 months.

Open source frameworks (Cayenne/Tapestry) catching up to WebObjects/Wonder

drop off in client requests. I expect if we decide to do SOA product, we may use it

The learning curve is too steep for us to justify training new developers, and we do not find many developers who know the technology. We have encountered several WO developers who think they know what they are doing but lack the skills to function effectively. Between the shortage of qualified developers and the expense required to train a new developer, the technology is no longer viable for us.

compatibility, performance

There is pressure to use other frameworks (such as Play).

I am primarily a front end designer and gave web objects secondary attention. I was knowledgeable enough to build a simple CMS application with a small eCom component, but I was always running into brick walls and could not find support when problems arose. Now much better tools are available free. I could no longer charge for development time.

Lack of support by Apple and community getting smaller.