Take full advantage of IBM's IDEs for end-to-end mobile development

ABSTRACT

Mobile development with Rational Application Developer 8.5, Rational Software Architect 8.5, Rational Developer for zEnterprise 8.5, Rational Developer for Power Systems 8.5 and IBM Worklight Studio 5

Table of Contents

Executive summary

IBM Worklight Studio 5 (IWS) is a new development environment for building mobile applications based on the IBM Worklight mobile platform. Rational Application Developer 8.5 (RAD) is the latest version of the award-winning IDE for application development on the IBM WebSphere platform. They each have their own unique strengths, and, when used together, create a comprehensive tool for developing enterprise solutions with a multi-channel front-end.

Rational Application Developer 8.5 has mobile web development tools targeting a pure web deployment solution.

- Developing applications, including mobile web applications, and deploying to WebSphere Application Server or WebSphere Portal
- Programming models include Java EE, OSGi, SCA and Web 2.0
- Testing and debugging applications on WebSphere or in IBM Smart Cloud
- Developing virtual application patterns for IBM Workload Deployer

Rational Software Architect for WebSphere Software 8.5 (RSA) is a superset of RAD. Any discussions about RAD 8.5 in this document is equally applicable to RSA 8.5.

IBM Worklight Studio 5 has tools for hybrid development within a multi-channel architecture.

- Developing applications and deploying to smart phones and tablets
- Programming model is HTML5 and JavaScript
- Multi-channel architecture covering mobile devices, mobile web, desktop web and desktop widgets

The two products can be installed together into the same Eclipse shell to get a comprehensive environment for multi-tier enterprise application development. Use the RAD tools for scalable middle tiers and robust back-ends. Use IWS tools for multi-channel or mobile-focused front-end.

IBM Rational® Developer for Power Systems Software™ (RD Power) provides rich desktop Integrated Development, Porting, and Optimization Environment for multi-platform development.

- Modern development environment for the major development workloads on IBM i, AIX, and Linux supporting remote file management, searching, editing, refactoring, application analysis, code coverage, performance, porting and migration advisors, build, and debug.
- Combines code analysis and profiling with performance management for expert guidance about improving application performance

RD Power and RAD can be installed into a common Eclipse shell to get a comprehensive environment for multi-tier mixed workload enterprise application development for Power Systems.

RD Power should not be installed into a common Eclipse shell with IWS because RD Power does not support the Oracle Java Runtime Environments (JRE). We recommend keeping IWS separate from RD Power.

IBM Rational Developer for zEnterprise (RD zEnterprise) 8.5 is a unified development tool for heterogeneous application development spanning systems such as z/OS, CICS, IMS, DB2, WAS and Power systems.

- Accelerates the development of your Mainframe applications, including traditional COBOL, PL/I, C/C++, EGL, Assembler, and Java/JEE applications, Web services, and XML-based interfaces across hardware platforms such as System z, Linux for System z, Power, and x86.

RD zEnterprise includes RAD, RDz, and RD Power. These IDEs can be installed together to have all the tools within a single workbench. RD zEnterprise also includes IWS. RDz, like RD Power as discussed above, should not be installed together into a common Eclipse shell with IWS.

Developing Mobile Web Applications with RAD

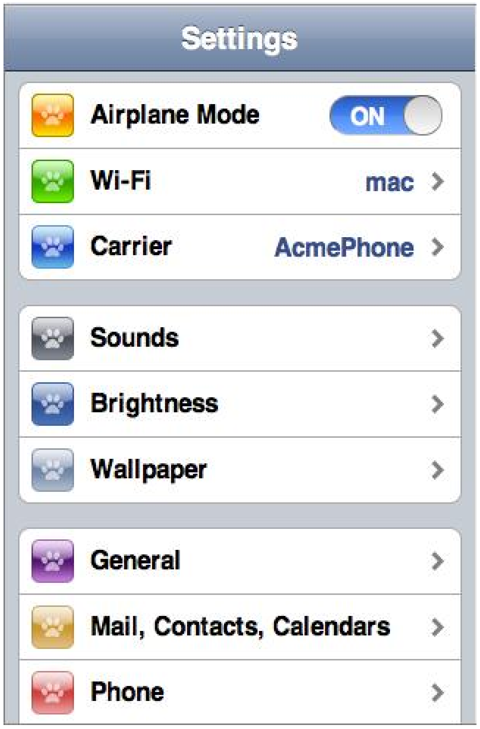
The browsers on today's mobile devices are powerful and able to render almost any existing web site written for the desktop browsers. However, because of their smaller display sizes, and different interaction gestures (touch/swipe/pinch vs. mouse click), it's almost always desirable to develop custom web sites targeting mobile browsers.

Responsive design is a popular approach to handle screen sizes and adjust layout and contents. It makes it possible to use a single code base to address a range of form factors.

However many mobile web sites are well beyond a different layout when compared with their desktop siblings. Consider a dashboard type of web application. The desktop version is typically laid out in a multi-column format so all the sub-views are visible at once. The mobile version, however, shows only one sub-view at a time, which takes up the whole screen, with transitions to go from one sub-view to another. Such a mobile application requires custom development where the implementation diverges from the desktop version quite significantly.

The ultimate mobile web site design is one that conforms to the UI design guidelines of the mobile platform itself, making it intuitive to use because of its familiar look-and-feel and the ability to respond to mobile gestures such as tap, swipe and pinch.

Using a JavaScript library such as Dojo Mobile or jQuery Mobile, a native looking mobile UI can be developed for the mobile browser environment.



Airplane Mode ON

Wi-Fi mac

Carrier AcmePhone

A native-looking iPhone web page written in Dojo Mobile

Rational Application Developer 8.5 supports developing web UI with HTML5 and Dojo Toolkit, including Dojo Mobile. These applications are deployed to a web server or an application server, such as WebSphere Application Server. Using Rational Application Developer 8.5, you can develop web UI using a range of techniques mentioned above, targeting both desktop and mobile browsers. This environment has excellent support for Dojo Toolkit, including Dojo Mobile. You can read more about these capabilities in the RAD product wiki (https://www.ibm.com/developerworks/wikis/display/rad/Home).

Developing Mobile Hybrid Applications with IWS

While web-based mobile applications are capable of achieving native look-and-feel, they are limited to functionalities accessible from inside the browser. If there is a need to take advantage of the device capabilities such as camera, accelerometer, or access to the contacts list etc., the application must run directly on the mobile platform.

To develop these so-called “native mobile applications”, you can use the platform vendor’s SDK and developer tools. But before you go in this “pure native” direction, you need to take the following factors into consideration:

- Do your IT teams have the necessary skills? Android and Blackberry platform APIs are in Java, iOS is Objective-C and Windows Phone 7 is C#. While Java and C# skills are relatively easy to find in today’s enterprise IT teams, Objective-C has only surfaced recently in the development community due to the rise of the iOS platform.

- How many platforms do you need to support? Going "pure native" means each mobile application has to be developed once for each target platform. This may not be economical with today's tight IT budget.

A cross-platform solution can help address these issues. IBM Worklight Studio offers an elegant solution to develop cross-platform mobile native applications. It's built on the concept of "mobile hybrid applications", which uses web technologies (HTML5, CSS, and JavaScript) as the primary programming languages. Using a JavaScript-to-native bridge called Apache Cordova, formerly known as PhoneGap, hybrid applications can access device capabilities without having to write in native platform languages.

On top of the unified programming model, IWS also maximizes code re-use across platforms with its unique optimization mechanism. You can read more about it in the Worklight user documentation (http://www.ibm.com/software/mobile-solutions/worklight/library/).

Multi-channel development with IWS

So far the decision tree is pretty clear-cut: for mobile web applications use RAD, for mobile hybrid applications use IWS. But what if your current IT project needs to address both of these channels? A typical scenario is that you want a mobile web site for maximum discoverability, because your existing customers already know the URL from their past experience in the desktop browsers; on the other hand, you want a mobile hybrid application for deep user experiences by taking full advantage of the device's capabilities. Finally the two channels should have similar look-and-feel for usability reasons.

So it often makes sense to use a single code base to enable these multiple channels. IBM Worklight Studio has multi-channel development at the front and center of its architecture. Worklight Studio projects support not only the popular mobile phone and tablet platforms, but also desktop web and widget systems. Thanks to its code layering and optimization mechanism, all these different channels are enabled by the same code base, and adding another channel is only a small increment on top of the existing application code. You can read more about the supported "environments", Worklight's terminology for channels, in the Worklight user documentation (http://www.ibm.com/software/mobile-solutions/worklight/library/).

Shell-sharing between RAD and IWS

If you have a need to use both tools for the same project, there are good reasons to install them into the same Eclipse "shell", so that you can do all the development activities in the same Eclipse workbench. It's not atypical that the same developers are responsible for both mobile UI and REST services, or other components on the server side like EJBs or Web Services. Furthermore, many IT groups find it easier and more economical to have a common development workbench for all developers in the organization.

Note, however, that currently there are certain limitations to shell-sharing between RAD 8.5 and IWS 5.0.

Because certain tools in RAD require an IBM Java Runtime Environment (JRE), you must launch the Eclipse workbench with an IBM JRE if you need to use those tools. Here's a list of tools in RAD that require an IBM JRE:

- Deploying, testing or debugging applications on WebSphere Application Server or WebSphere Portal
- Profiling an application
- Running code coverage
- Developing J2C Connectors for CICS, COBOL and PL/I systems
- Developing portlets
- Developing JAX-WS or JAX-RPC services

To launch with an IBM JRE, use the launch point from the system Start menu called "IBM Software Delivery Platform -> IBM Rational Application Developer 8.5 -> Rational Application Developer".

On the other hand, if Android is a target platform for the mobile application development, you must install Android Development Tools (ADT) into the RAD-IWS workbench in order to compile, package and test the Android applications produced by IBM Worklight Studio. ADT currently requires an Oracle JRE to perform the code signing operation during Android builds. This means the RAD-IWS-ADT workbench must be launched with an Oracle JRE.

Another common scenario that benefits from launching with an Oracle JRE is developing applications for WebSphere Application Server (WAS) 8.5 Liberty profile. The Liberty profile runs with any JRE 1.6 or later. It provides a high-fidelity WebSphere development environment with many developer-friendly features such as lightning-fast startup time and instantaneous processing of frequent application artifact updates. It is a more desirable development target than Tomcat or other open-source application servers, because using the Liberty profile guarantees that the application is fully compatible with the full WebSphere profile. Using the Liberty profile as the development target in the RAD-IWS-ADT workbench, and launching with an Oracle JRE, provides a comprehensive development environment.

To launch with an Oracle JRE, use the launch point called "IBM Software Delivery Platform -> IBM Worklight Studio 5.0 -> IBM Worklight Studio".

For other development tasks that do not require a particular JRE, either launch point can be used.

Shell-sharing between WebSphere Application Server Developer Tools and IWS

WebSphere Application Server Developer Tools for Eclipse (WDT) is a new offering as part of the RAD 8.5 and WebSphere 8.5 releases. WDT is a subset of RAD, and has programming model support for JEE, OSGi, JAX-RS and Web. WDT is available at no cost and is unsupported. Customers can buy support if so desired. It also supports running with an IBM or Oracle JRE.

Because WDT doesn’t include many of the advanced tools in RAD that require an IBM JRE to run, the only scenario that requires an IBM JRE to launch WDT is if the development target is one of the full WebSphere Application Servers supported by WDT. However, you should consider using WebSphere Liberty profile instead for reasons mentioned in the previous section. The WebSphere Liberty profile runs on any JRE 1.6 or later, including IBM and Oracle JREs.

WDT and IWS can be installed together. Add Android Development Tools and the WebSphere Liberty profile, and you get a comprehensive development environment supported on Oracle JREs.

Setting Up Development Environments on Mac

For developing hybrid or pure native iOS applications you need Xcode, Apple’s IDE for iOS and Mac platforms. Xcode is only available on Mac. As a result you will need Mac for your team in order to debug and package hybrid or native iOS applications.

Mac support is relatively new in the history of Rational IDEs. RAD 8.5 is currently only supported on Windows and Linux; however, WDT 8.5 and IWS 5.0 are supported on Mac as well. The WebSphere Liberty profile is also supported on Mac.

WDT can be installed together with IWS and the WAS Liberty profile, giving you an end-to-end development environment on the Mac.

Note that IBM Installation Manager is not currently supported on Mac. Installing WDT and IWS on Mac is done through Eclipse’s native install mechanism using update sites.

Developing iOS Hybrid Apps on Windows or Linux

Even for the team members with a Windows or Linux development environment, they can still contribute to iOS development. Using IWS they can work on the web resources, HTML/CSS/JavaScript, for either the common layer or the iOS-specific layers. To test the code in a native iOS environment, they can check in the code to the team repository and have a build system (such as Rational Team Concert) compile and package the installable binary, the .ipa file.

The build can then publish the .ipa file to the IBM Worklight Application Center, which is the private AppStore for development purposes. For a typical Worklight application it usually takes a minute or two for all the build steps to finish and for the new version of the application to appear in the AppCenter’s catalog. After performing a quick install or update from the iOS device, the code is ready for testing.

Read more about RTC build system integration and Application Center at http://www-01.ibm.com/software/mobile-solutions/worklight/library.

Rational Team Concert Client Integration

All of the IDEs mentioned above, RAD, WDT, IWS, can be installed together with Rational Team Concert (RTC) Eclipse client 3.0.1.3 and 4.0. Installing the RTC Eclipse client into the same Eclipse shell enables the world-class team development support by RTC. Both 3.0.1.3 and 4.0 support IBM and Oracle JREs. In the context of shell-sharing with IWS, the RTC 4.0 client is supported on Mac in addition to Windows and Linux. Read about RTC and other Rational Collaborative Lifecycle Management products at jazz.net.

Additional Resources

- RAD wiki: *https://www.ibm.com/developerworks/wikis/display/rad/Home*
- WAS Liberty Profile and WDT: *http://wasdev.net*
- RD zEnterprise: *http://www-01.ibm.com/software/rational/products/developer/zenterprise/*
- RDp: *http://www-01.ibm.com/software/rational/products/rdp/*
- Worklight Studio: *https://www.ibm.com/developerworks/mobile/index.html*
- Worklight Studio installation guide: *http://public.dhe.ibm.com/ibmdl/export/pub/software/mobile-solutions/worklight/docs/wl_5_0_dev_env_install.pdf*
- Twitter
  - RAD, WDT, WAS Liberty: @wasdevnet
  - Worklight and other IBM Mobile solutions: @ibmmobile