IBM Worklight V5

Technology overview

Contents

IBM Worklight—Overview

IBM Worklight software can enable organizations to extend their business through HTML5, hybrid and native applications (hereafter referred to as "apps") for a variety of smartphones and tablets, and to support the entire application lifecycle from development, through back-end integration and authentication, to post-deployment management.

Development: The IBM Worklight Studio and the IBM Worklight SDK simplify the development of web, hybrid and native apps across multiple mobile platforms, including iOS, Android, BlackBerry and Windows Phone. The IBM Worklight optimization framework can enable the delivery of a rich user experience that matches styling requirements of multiple target environments. In the process, IBM Worklight can maximize the sharing of the code-base from one environment to the other, effectively reducing costs of development, time to market and ongoing management efforts.

Integration: The server architecture and Adapter technology of IBM Worklight simplify the integration of apps with back-end enterprise systems and cloud-based services. The IBM Worklight Server is designed to seamlessly fit into the organization's IT infrastructure and leverage its existing resources wherever possible. The standalone back-end integration layer can be customized and shared among multiple applications. Furthermore, IBM Worklight Adapters support two types of data delivery mechanisms: device requests and push notifications.

Run time: The IBM Worklight Studio prepares application files for upload to the various public app stores and private distribution repositories. Running apps communicate with any enterprise back-end systems and cloud-based services through the IBM Worklight Server, which optimizes data for mobile delivery, supported by a variety of security features to protect sensitive user data.

Manage: Once deployed, administrators can monitor and control the access of different apps and devices to back-end systems; directly update and disable them based on predefined rules; and control all push services and event sources from one centralized web interface called the IBM Worklight Console. In addition, administrators can access usage information about the installed app base and its users, using built-in and customized reports.

IBM Worklight—Components

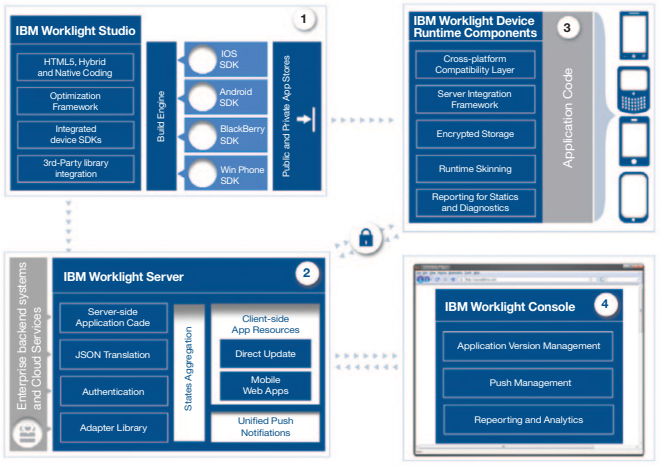
The IBM Worklight architecture consists of four main components:

- The IBM Worklight Studio—the platform's integrated development environment (IDE)
- The IBM Worklight Server—a gateway between apps, back-end systems and cloud services
- IBM Worklight Device Runtime components—complementing the server with client-side functions
- The IBM Worklight Console—a web-based administration interface

Development tools

The IBM Worklight Studio

The IBM Worklight Studio is an Eclipse-based IDE that allows developers to perform all the coding and integration tasks that are required to develop rich employee- and customer-facing applications. The IBM Worklight Studio augments the familiar tools of Eclipse with a wide variety of enterprise-grade features that are delivered by the IBM Worklight Plug-in, enabling it to streamline application development and facilitate enterprise connectivity.



The following are some of the main features that are supported by the IBM Worklight Studio:

Cross-platform support

The IBM Worklight Studio enables the development of rich hybrid and native mobile applications on iOS, Android, BlackBerry and Windows Phone smartphones and tablets. Mobile HTML web apps can be developed for a wider array of modern devices that are using web browsers capable of running HTML 4, CSS 2.1 and JavaScript 1.5 at a minimum.

Using its Optimization Framework, IBM Worklight differentiates itself from other technologies in the market that deliver a lowest-common-denominator solution, by enabling developers to share the majority of the application code across multiple environments, without compromising the user experience or the application functionality.

Using a unique file structure within the IBM Worklight Studio, the Optimization Framework enables developers to adjust applications to the different mobile environments, while maximizing the reuse of the code-base. Developers can share the common app code among multiple environments, while isolating environment-specific code in designated folders that can overwrite or augment the commonly shared code. As a result, application logic remains consistent among the different environments, while the user interface (UI) behaves natively and adheres to end-user expectations and the unique functionality and design guidelines of the device.

Application developers can directly access the application programming interfaces (APIs) that modern devices offer, and can more easily integrate publically available or customized third-party libraries, frameworks and tools, resulting in advanced mobile applications built according to the unique and specific needs of the organization.

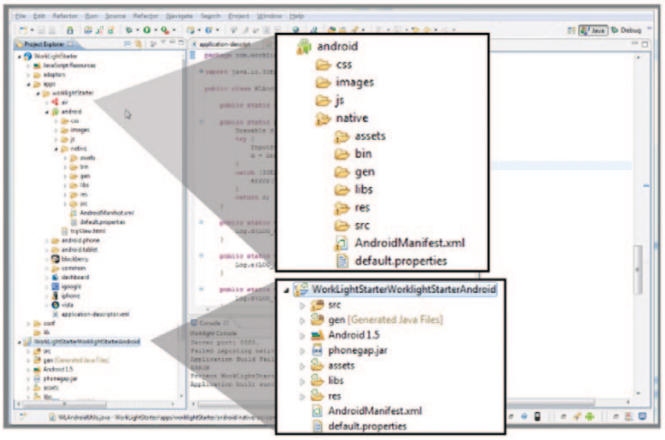
Because developers are not dependent on an intermediary build-time or runtime layer, such as a cross-compiler or interpreter, using IBM Worklight, native APIs are accessible upon release of new mobile OS versions or third-party libraries. Furthermore, the app's web code is executed directly by the mobile browser, so developers have direct access to the HTML Document Object Model (DOM) and are free to use any JavaScript API or third-party JavaScript toolkits and frameworks.

Hybrid coding

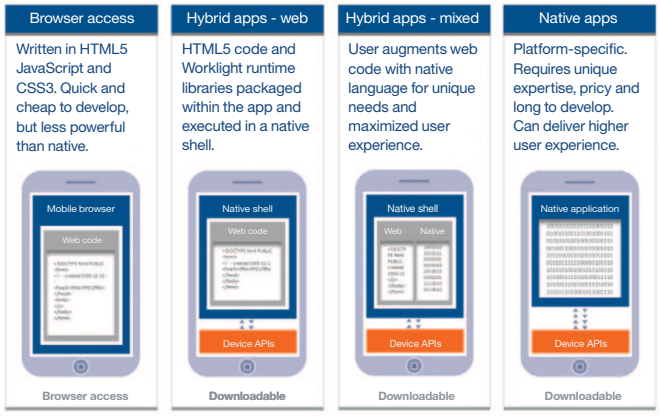
Facing the constantly evolving fragmented ecosystem of mobile devices and operating systems, application development has become a costly, yet unavoidable endeavor for business, today. This challenge has created a market for cross-platform mobile development solutions that is rapidly growing.

However, to achieve cross-platform capabilities, many solutions in the market rely on limiting proprietary tools, such as form-based IDEs, WYSIWYG (what you see is what you get) tools, or simply prepackaged apps. The result is an unavoidable tradeoff between user experience and multiplatform coverage. Using IBM Worklight, developers can choose between using pure native code (Objective-C, Java or C#), standard web technologies (HTML5, CSS3 and JavaScript) or a combination of both within the same app, enabling developers to strike the desired balance between development efficiency and app functionality.

Developers can call native code from HTML-based pages, combine HTML and native-based pages in the same application and display HTML and native components together on the same page.



No single development approach offers a complete solution to the larger challenge, but by using the unique support for hybrid coding provided by IBM Worklight, organizations are able to use the same mobile platform to develop, run and manage a variety of app types based on the specific needs of the project at hand.



Runtime skins

Further optimization of apps is possible within the IBM Worklight Studio by using runtime skins. These skins are packaged with the app's executable, applied to the mobile app during run time and enable it to automatically adjust to different devices from the same OS family. Common scenarios that benefit from runtime skins follow.

Support for HTML5

Support for HTML5 by IBM Worklight leverages a standards-based approach, enabling developers to write HTML5 code directly into the app without the limitation of proprietary interpreters or code translators, thereby benefiting from capabilities such as:

- A cleaner, more readable and consistent HTML code.
- Access to rich media types (audio and video), available previously by way of native code only.
- Use of advanced UI components, such as data pickers, sliders, edit boxes that automatically support ellipsis and others—implemented natively by the browser.
- Use of Cascading Style Sheets 3 (CSS3) styles and CSS3-based animation to reduce app size and improve its responsiveness.

- App distribution channels that go beyond the different app stores and their time-consuming and limiting restrictions.
- Support for geolocation services.
- Offline storage capabilities.

IBM Worklight further augments these capabilities with enterprise-grade utilities, such as on-device encryption and offline user authentication to increase application availability.

Support for third-party JavaScript toolkits and UI frameworks
On top of its support for HTML5, the IBM Worklight platform provides seamless integration with UI frameworks, such as jQuery Mobile, Sencha Touch and dojox.mobile. Developers can pick the JavaScript UI framework of their choice and use it to develop their application within the IBM Worklight Studio.

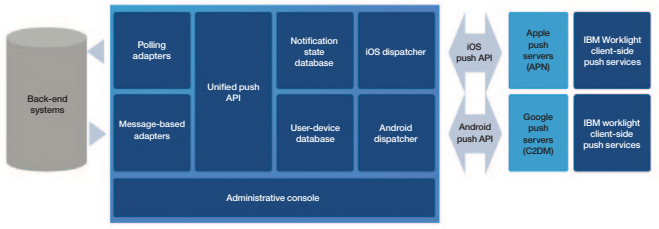
Native-device SDK integration
The IBM Worklight Studio tightly integrates with the software development kits (SDKs) of the mobile devices that IBM Worklight supports. This enables developers to take full advantage of the native code capabilities and the best-in-class development tools, testing and debugging mechanisms that are native to the mobile device, without leaving the development environment.

Standardized data retrieval
The IBM Worklight Studio enables developers to use XSL transformations and JavaScript code to convert retrieved hierarchical data from any back-end system to JavaScript Object Notation (JSON) format, thus preparing it for app consumption. Developers can invoke back-end services directly from within the IBM Worklight Studio and receive raw results in Extensible Markup Language (XML), or processed results (after having converted to JSON using Extensible Stylesheet Language (XSL) transformations and JavaScript) in JSON format.

Developers can perform server-side mashups in JavaScript to collect data from various back-end applications and streamline them to the device, thereby reducing the number of requests made on the slow mobile network and greatly improving app responsiveness.

In addition, developers can choose to implement server-side back-end integration and authentication code in Java, rather than in JavaScript.

Unified push notifications
In the process of creating the integration adapters, users can use the uniform push architecture of IBM Worklight to preconfigure automatic alerts from one centralized interface. Leveraging its unified push API for its supported devices, IBM Worklight makes the entire process of communicating with the mobile vendor completely transparent to the developer.



Distributed development

Enterprise mobile development is rarely a simple process conducted by one developer. Most commonly, the complex enterprise development environment consists of multiple development, testing and QA teams all working on different portions of the app, sometimes even from different geographical locations. IBM Worklight is designed to support such scenarios through a variety of features and functions, including integration with other IBM collaboration tools.

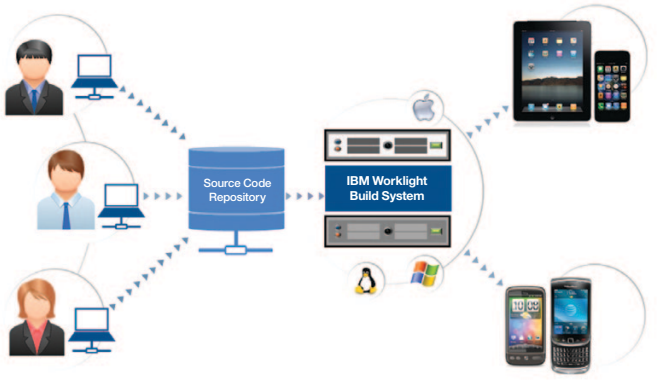
Centralized build

The IBM Worklight Builder is a stand-alone application that can be more easily integrated with common central build services, such as IBM® Rational® Jazz™ Builder, Hudson and Luntbuild. Leveraging the centralized build functionality, the different teams involved in the development, testing and QA phases can work off of one common version of the code, effectively enhancing the collaboration and automation of the internal application development process.

The shell approach

Most often, enterprises employ multiple development teams with different skills and expertise. The "shell" approach enables such companies to reduce the internal barriers of mobile development, making it ubiquitous throughout the organization by compartmentalizing skill sets and responsibilities.

The shell approach of IBM Worklight breaks down the development of the app into two portions: an external shell and an inner application.



The shell consists of a customizable container that provides JavaScript access to the native capabilities of the device. A devoted team of expert developers are responsible for its branding, security configurations, audits and authentication frameworks. The team can create a variety of shells, each carrying different policies and branding, forcing inner apps running within each shell to automatically comply with its parameters. Such parameters could include restriction of access to data, use of certain APIs, different branding and so forth.

With the corporate policies enforced by the shell, the inner apps can be more easily built by departmental development teams, using nothing but web languages. Such teams are only required to focus on the user interface, the business logic and, potentially, data integration. Distribution of the app or apps can be achieved by way of three different channels:

- An inner app can be fused into a shell by the centralized build server and uploaded to a private or public app store, while new versions of the inner app are sent to and updated directly (subject to the vendor's terms of service) on the end-user device.
- A shell can be packaged with a directory of corporate-sanctioned applications, enabling users to choose a different inner app according to their needs.
- A shell can be distributed empty to the user, who will then access a repository of applications stored on the server.

The IBM Worklight SDK

The IBM Worklight SDK is a set of libraries and tools that support the development process and simplify integration of client-side code with back-end systems, cloud-based services and authentication mechanisms. The IBM Worklight SDK consists of:

- JavaScript client API, providing access to IBM Worklight services, environment-specific features and a set of cross-platform UI utilities.
- Objective-C client API, providing access to IBM Worklight services for iOS apps that were developed using Objective-C.
- Java client API, providing access to IBM Worklight services for Android OS applications that were developed using native Java code.

- JavaScript server API, enabling preprocessing of requests from applications before forwarding them to back-end systems, post-processing of back-end data before sending it to applications and mashup of back-end data from multiple sources to reduce traffic loads and latency on the mobile network, thus improving app responsiveness.
- Offline access API, allowing developers to specify access failure routines, check connectivity state during run time and automatically re-establish connectivity.
- Server- and client-side unified push API that support the dispatching of notifications, registering to push services and providing unified management.
- Localization API to detect the locale of the mobile device.
- Validating schemas for adapter configuration.
- Full documentation of the development and integration processes and of all APIs.
- Complete training material, including presentations, exercises and sample code.

Runtime server environment

The IBM Worklight Server

The Java-based IBM Worklilght Server is a scalable gateway between apps, external services and the enterprise. The IBM Worklight Server helps facilitate encrypted communication, back-end connectivity, data manipulation, authentication, analytics and operational management functions supported by a variety of security features. The IBM Worklight Server can:

- Provide adapter technology that connects to a variety of enterprise information systems over widely used integration technologies, such as Simple Object Access Protocol (SOAP), representational state transfer (REST), Structured Query Language (SQL), Lightweight Directory Access Protocol (LDAP), SAP and more, as well as cloud-based services.
- Enable multisource data mashups to efficiently integrate several data streams into one and serve it to the application user. Multisource data mashups are not only an effective way of servicing the right data to the user, but also reduce overall traffic in the system.
- Enable developers to add custom server-side logic necessary for delivering back-end data for mobile consumption. This not only helps distribute processes between the client and server, but also helps address data-security regulations within the organization.
- Integrate with the corporate authentication infrastructure to help secure application and data access, in addition to transaction invocation. The IBM Worklight authentication infrastructure is flexible enough to support different types of authentication—from multifactor or multistep login processes to non-interactive single sign-on (SSO) integration, in addition to offline authentication of users to increase app availability. Furthermore, the IBM Worklight Server simplifies the integration with HTTP-based services that require authentication. Integration with Kerberos, Windows NT LAN Manager (NTLM), Basic and Digest authentication can be more easily achieved by simple configuration of the HTTP adapter, without having to write server-side code. The server also supports device-based application SSO, enabling apps to be automatically authenticated if an existing authenticated session is already available through the same mobile device.
- Employ standard and proprietary security mechanisms to help prevent attacks (more about security).
- More easily scale to support hundreds of thousands of users and multiple applications through physical clustering.
- Provide app-deployment and version-control features that are managed and accessed by the IBM Worklight Console.

- Be integrated with IT monitoring and performance management systems that verify the vitality of the IBM Worklight Server and the services it provides to applications.
- Automatically collect user-adoption and usage data for auditing and reporting purposes and allow for the custom configuration of reporting metrics. Raw data can be more easily exported for further analysis by the different business intelligence tools used by the organization.

The IBM Worklight Console

The IBM Worklight Console is a web-based user interface dedicated for the ongoing administration of the IBM Worklight Server and its deployed apps, adapters and push-notification services. Through the console, administrators can:

- Access administrative dashboards that monitor all deployed adapters and applications.
- Control and monitor all push-notification services, event sources and related applications.
- Manage multiple versions of the same application and remotely disable applications by version and mobile-operating-system type.
- Access built-in reports of application adoption and usage.
- Define device-based access-control policies to control access of apps.

IBM Worklight Device Runtime components

IBM Worklight provides client-side runtime code that services HTML5, hybrid or native apps. Capabilities include:

- **Access back-end data and transactions**: API for the invocation of IBM Worklight services, retrieval of data and execution of transactions against back-end systems.
- **Authentication and security**: API and code for managing the authentication sequence and securing the application data and its link to the IBM Worklight Server.
- **Application Management**: API and code for applying new application versions and disabling applications in accordance with policies defined in the IBM Worklight Console.
- **Troubleshooting**: Code for detecting runtime connectivity problems in the app and collecting troubleshooting information about the app and the device.
- **Usage reporting for audit and analytics**: API for collecting built-in and custom data from apps, to be recorded by the IBM Worklight Server for audit and analytics purposes.
- **Cross-platform compatibility APIs**: Uniform API for device features and useful UI tasks, hiding the differences across different environments.
- **Application of skins**: Enables developers to adjust the features and functions of the app to the device's form factor in run time, optimizing it for different versions of the same device family.

The runtime client environment consists of the following components:

- **JavaScript libraries:** A set of JavaScript libraries implementing the JavaScript APIs. These libraries are available in most runtime environments (with the exception of native iPhone and Android apps, which are written in Objective-C and Java, respectively, and which do not require JavaScript libraries).

- **Native libraries for hybrid apps:** A set of native libraries (for iOS and Android) that provide access to device-specific features. Apps written in JavaScript do not access these libraries directly, but rather through the relevant JavaScript APIs. In some cases, native code runs the web code provided by the developer.
- **Native libraries for native apps:** A set of native libraries for iOS and Android that provide access to IBM Worklight Server functionality for natively written apps.
- **Native code templates:** For iOS, Android, BlackBerry and Windows Phone devices, native-code templates encapsulating a browser that runs the web code provided by the developer.

Security and authentication mechanisms

IBM Worklight provides multiple mechanisms and tools that help support the creation of secure applications.

The following is a list of the main security features of the platform:

Mechanism	Benefit	Details
On-device encrypted storage	Protect sensitive information from malware attacks and device theft	• Uses AES256 and PCKS #5-generated encryption keys for storing app-generated information on the device • Allows offline user authentication • Implemented in JavaScript (highly obfuscated) with optional native performance enhancements
Direct update	Ensure timely propagation of updated app versions to the entire install base	• New versions of the code can be distributed without requiring the manual update of the app (applicable to web resources)
Remote disable	Enforce timely adoption of critical security updates to the entire install base	• Server-side console allows configuration of allowed app versions. Administrator can force users to install security updates to the native code
Authentication framework	Reduce overall cost and complexity of integration with authentication infrastructure	• Server-side architecture designed for integration with back-end authentication infrastructure based on Java Authentication and Authorization Service (JAAS) concepts, with authentication realms • Ready-to-implement integration with Kerberos, NTLM, Basic and Digest authentication • Ability to encrypt server-to-server SOAP communication with X509 certificates, following the Web Services Security (WSS) standard • Client-side framework for asynchronous login requests on session expiration

Mechanism	Benefit	Details
Server-side safeguards	Prevent SQL Injection and protect against cross-site request forgery (XSRF)	• Prepared-statement enforcement • Validation of submitted data against session cookie
Enterprise SSO integration	Leverage existing enterprise authentication facilities and user credentials and enable employee-owned devices	• Client-side mechanism obtains and encrypts user credentials, sends to the server with requests • Encryption incorporates user-supplied PIN, server-side secret and device ID • Credentials cannot be retrieved from lost or stolen device
Virtual private network (VPN) alternative	Enable delivery and operation of mobile apps for employee-owned devices or device types not allowed on the corporate network, and enable delivery when installation of VPN client on mobile devices is not possible or is complicated to manage	• Client-side and server-side frameworks act as secure socket layer (SSL)-based VPN • Network access control and policies preconfigured in the client-side framework layer • Network access and security measures updated using server-side framework • On-device encrypted storage to prevent compromise of sensitive data

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside a client's enterprise. Improper access can result in information being altered, destroyed or misappropriated, or can result in misuse of systems to attack others. Without a comprehensive approach to security, no IT system or product should be considered completely secure and no single product or security measure can be completely effective in preventing improper access. IBM Worklight systems and products are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures and may require other systems, products or services to be most effective. IBM Worklight does not warrant that systems and products are immune from the malicious or illegal conduct of any party.

System requirements

Production environment

The IBM Worklight Server can be installed on the following operating systems:

- Windows Server 2008 64 bit
- Red Hat Enterprise Linux (REHL) Version 4 SP3 or later (SP4 recommended) 64 bit

The server requires the following databases to store metadata and cached back-end data:

- My SQL 5.0.22 or later (5.0.67 recommended)
- Oracle 10g and 11g

The IBM Worklight Server can run on the following application servers:

- Tomcat 7
- IBM WebSphere® Application Server 7 and higher over Linux

The IBM Worklight Server can be clustered to achieve high availability and scalability. In such case, a load balancer is required. This can be any commercial load balancer, software or hardware, which supports sticky sessions. The load balancer can optionally act as a reverse proxy and SSL accelerator.

Development environment

The IBM Worklight development environment includes the IBM Worklight Server and the Eclipse-based Studio. The development environment is supported on the following operating systems:

- Windows 7, Vista or XP (32 or 64 bit)
- Macintosh environment

For development purposes, the following database is supported:

- My SQL 5.0.22 or later (5.0.67 recommended)

The IBM Worklight Studio requires the following distribution of Eclipse:

- Eclipse for Java Platform, Enterprise Edition (Java EE) developers, version Indigo

IBM Worklight is part of the IBM Mobile Foundation family of products that provides all the essential elements needed for complete mobile development, deployment and management within a business. IBM Mobile Foundation consists of capabilities that enable:

- Mobile application development and delivery—IBM Worklight.
- Complete end-to-end Mobile Device Management (MDM)—IBM Tivoli® Endpoint Manager.
- Advanced connectivity to back-end systems and cloud-based services that is optimized for mobile devices—IBM WebSphere Cast Iron® Cloud Integration.

For more information, please visit the following website:
ibm.com/software/mobile-solutions/worklight

For more information

To learn more about IBM Worklight assets for mobile application development, please contact your IBM marketing representative or IBM Business Partner, or visit the following website: **ibm.com**/software/solutions/mobile-enterprise

Additionally, IBM Global Financing can help you acquire the software capabilities that your business needs in the most cost-effective and strategic way possible. We'll partner with credit-qualified clients to customize a financing solution to suit your business and development goals, enable effective cash management, and improve your total cost of ownership. Fund your critical IT investment and propel your business forward with IBM Global Financing. For more information, visit:
ibm.com/financing

IBM Corporation
Software Group
Route 100
Somers, NY 10589

Produced in the United States of America
April 2012

Please Recycle

WSW14181-USEN-01