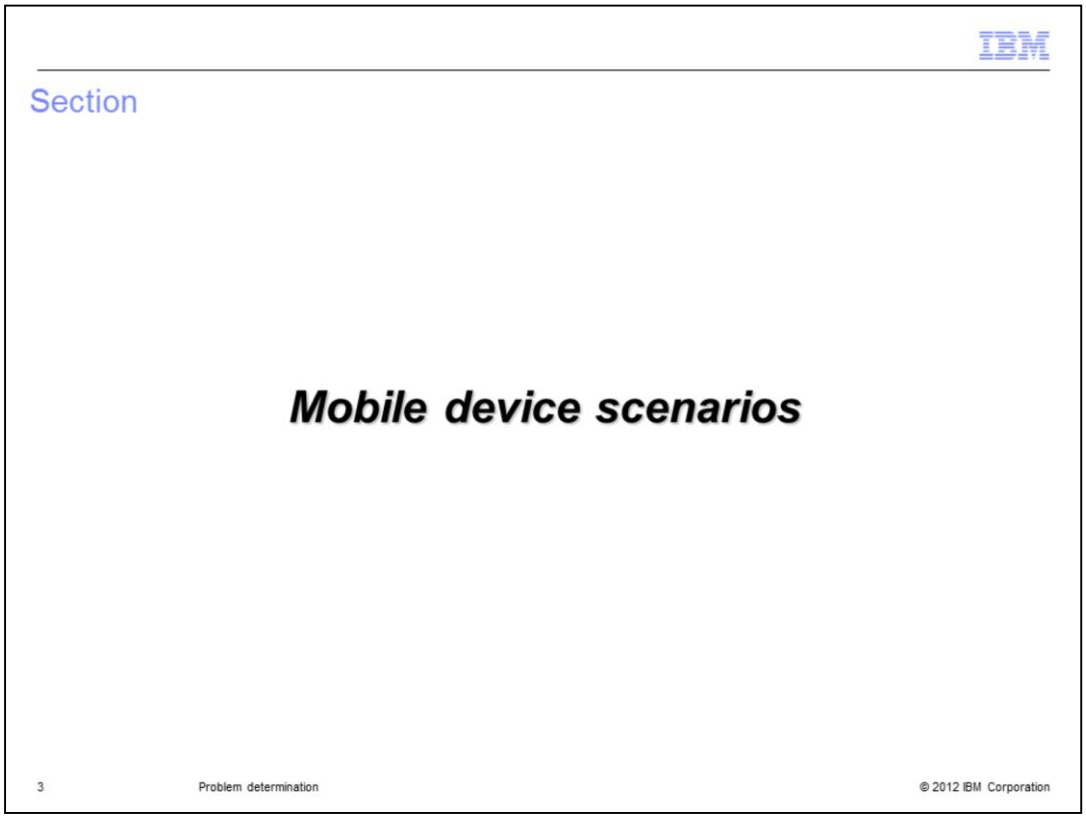
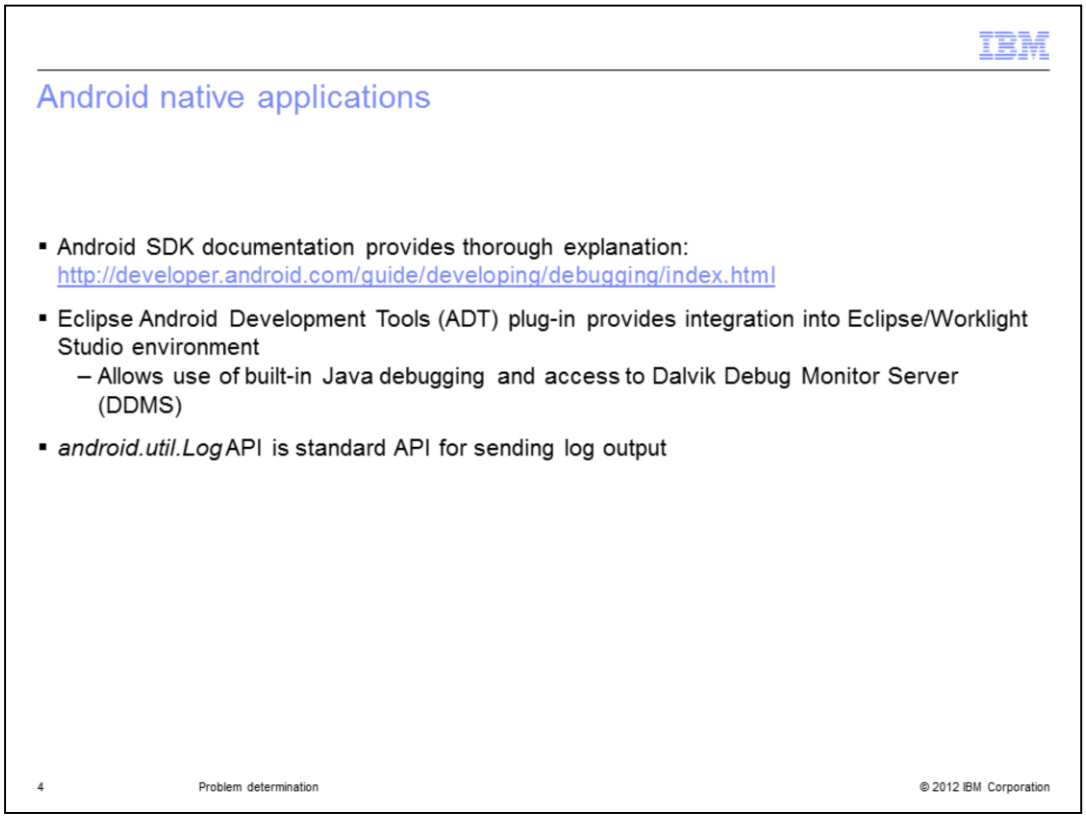
This module will provide an overview of problem determination in an IBM Worklight and Mobile Foundation environment.

The IBM Worklight and Mobile Foundation products involve interactions across a wide range of platforms. When building and debugging an application you can expect to frequently cross the boundaries between mobile devices, such as an iPhone running iOS or an tablet running Android, the mobile framework server which provides the supporting infrastructure for the mobile application, and legacy business services.

Because each of these environments has different capabilities and serves a different purpose, they have individually evolved their own patterns and tools for performing problem determination. This presentation will provide an overview of the mobile device and framework server portions of problem determination, and attempt to provide a foundation for understanding how to debug issues.

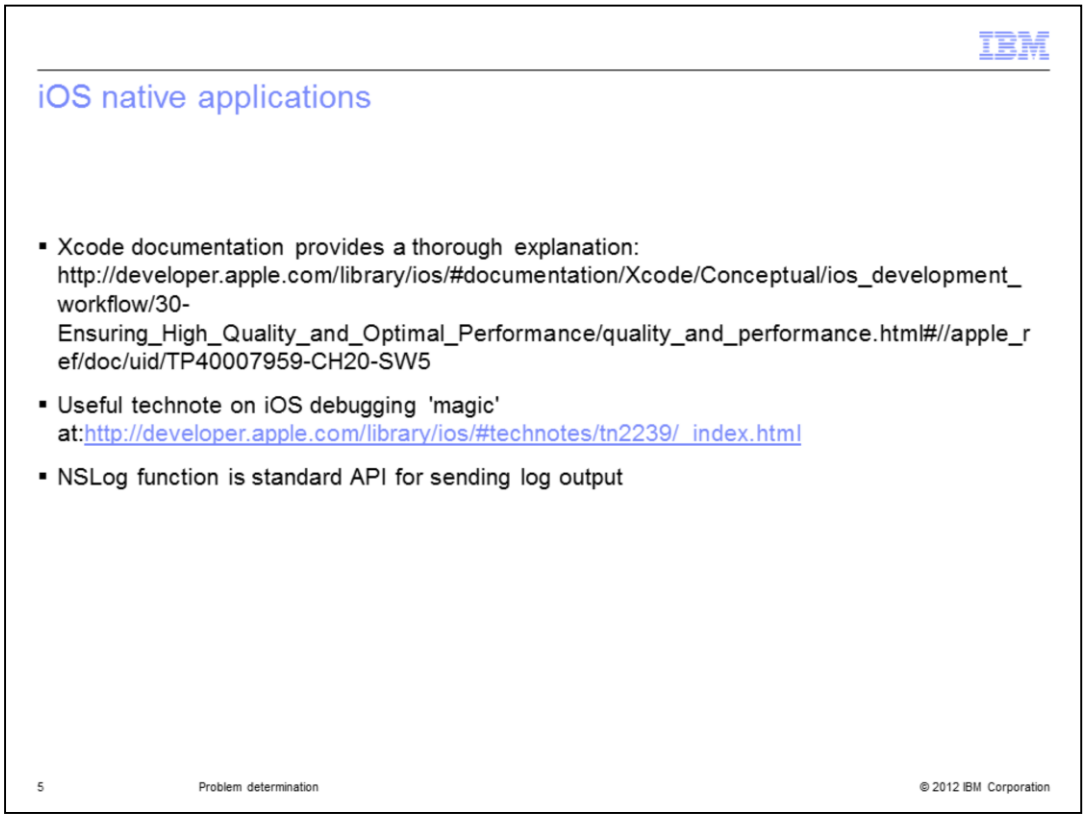


The following scenarios provide an introduction to the problem determination and debugging facilities available on mobile devices.



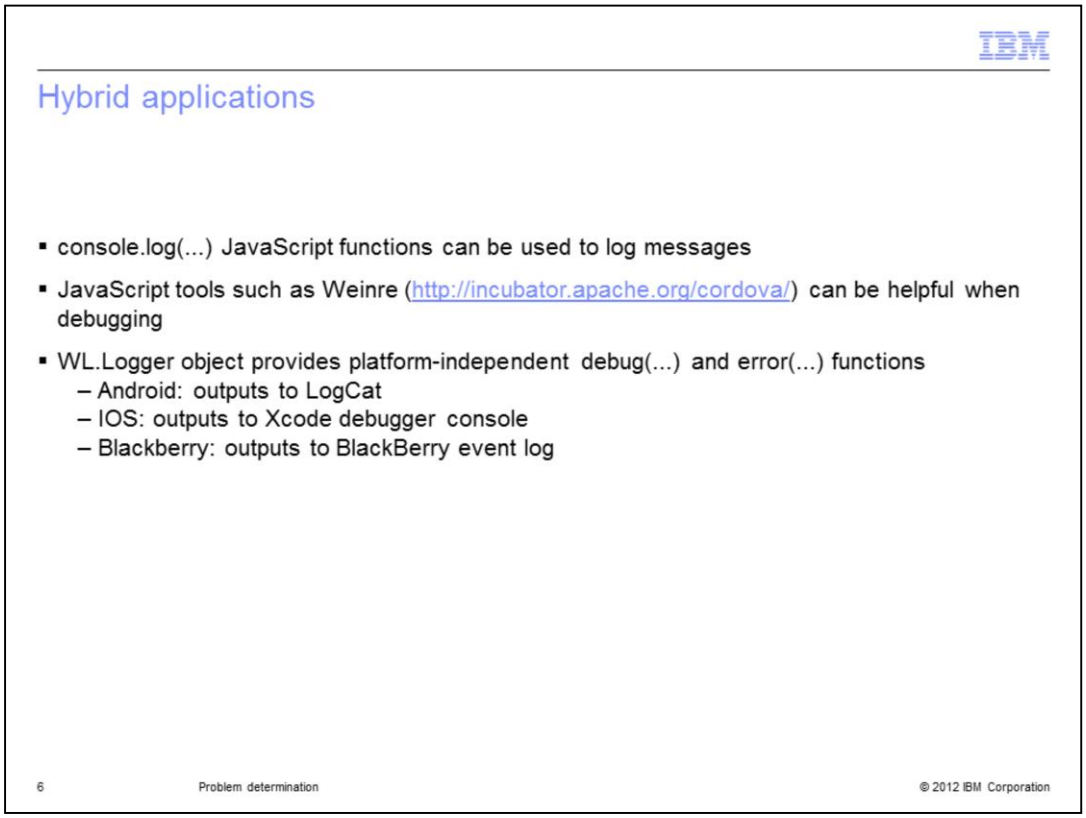
The Android SDK documentation provides a good resource to aid in the understanding of debugging native Android applications. The Eclipse Android Development Tools (ADT) plug-in provides an integration of the Android SDK tools into the Eclipse/Worklight environment, and allows access to the Dalvik Debug Monitor Server (DDMS) and a link any connected Android device so that logs and other information can be monitored.

The Android logging mechanism is available through standard native APIs and is quite useful during debug sessions. Access to logs is provided through DDMS by way of the logcat command. If using the ADT plug-in in Eclipse, the log messages will also appear in the Console view. One word of warning: the Android logs are fixed size and will wrap, thus potentially erasing valuable information for your debugging session if you do not retrieve it in time.



The Xcode documentation provides a good resource to aid in the understanding of debugging native iOS applications. Although Eclipse does not integrate the with Xcode development environment, the Xcode environment itself provides a comprehensive set of tools for aiding in the debugging of iOS applications.

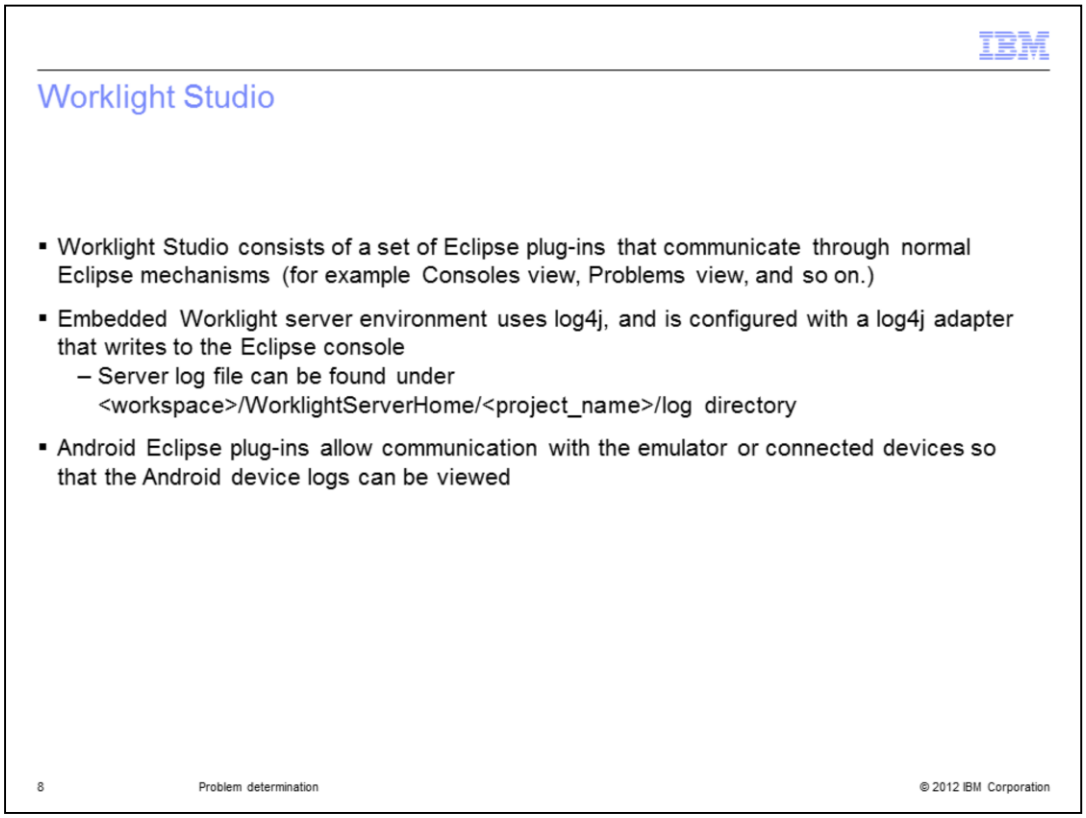
The iOS logging mechanism is available through standard native APIs and is quite useful during debug sessions. Access to the logs is made available through the Xcode environment.



As hybrid applications run through the web container on the target platform, you have a combination of tools available to assist in debugging. The JavaScript debugging and alert APIs can be used to log information or to create popups on the device screen. JavaScript debugging tools, such as Weinre which is part of the Apache Cordova project, can also provide valuable assistance in many situations. Native tools may also be used, although some of the information may be obscured.

The Worklight API itself provides a valuable Logger object which sends the log messages to the appropriate location for the executing platform. Native mechanisms for retrieving the log information can then be used to access the information as needed.

The following scenario provides an introduction to problem determination and debugging in the Workplace Studio development environment.

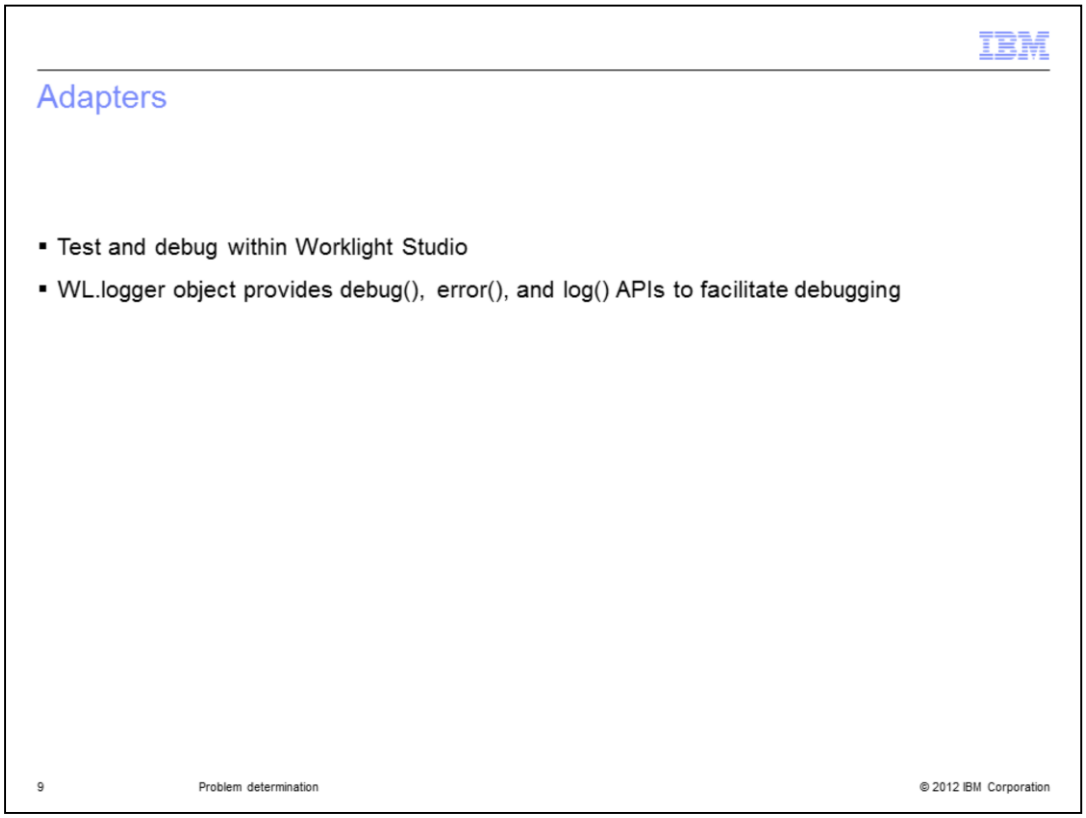


Worklight Studio consists of a set of plug-ins that are integrated into the Eclipse environment, and therefore the tools and experience should be familiar to those who have developed in the environment before. Errors and log messages are reported through the standard Eclipse views.

Studio also contains an embedded Worklight server environment into which applications can be deployed and tested. The embedded environment logs information to the Eclipse console and maintains a log file in the workspace.

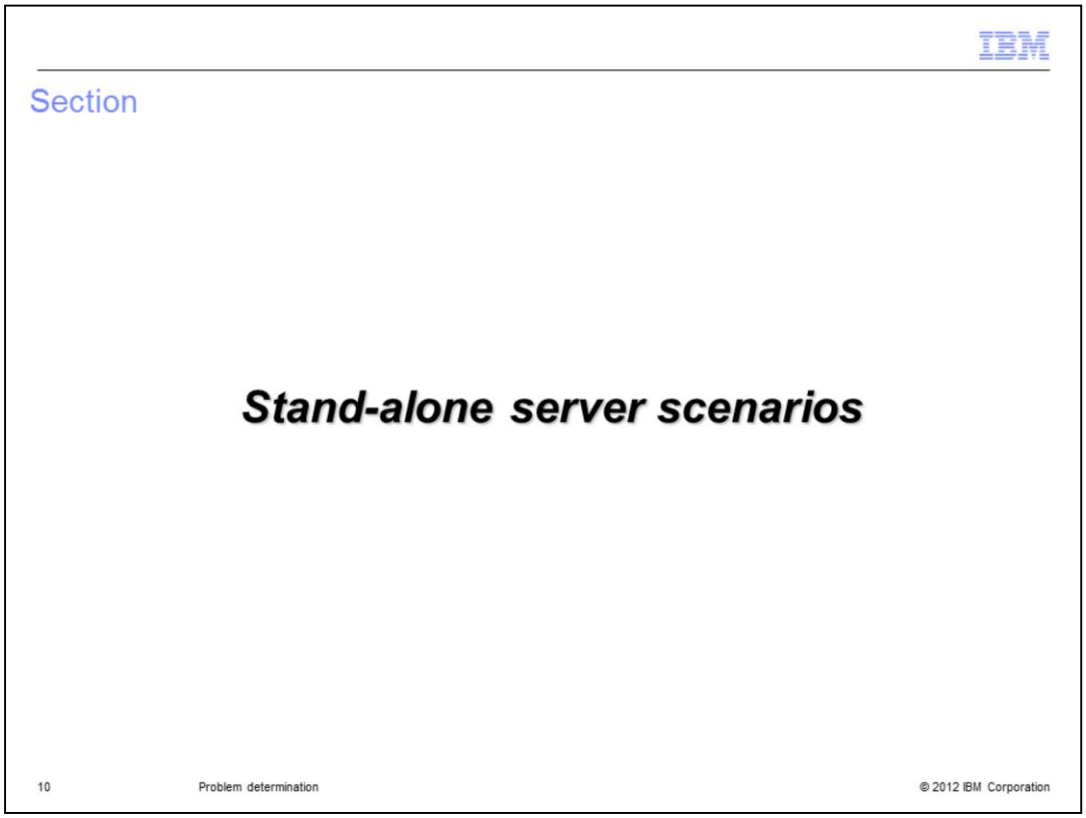
Most problems within the development environment can be resolved by looking at the Problems view or at one of the various Console views.

As was mentioned in the Android native section, the Android Eclipse plug-ins also allow the Android devleopment environment to be more thoroughly integrated with Studio, thereby providing more direct access to an application running on the emulator or on a connected device.

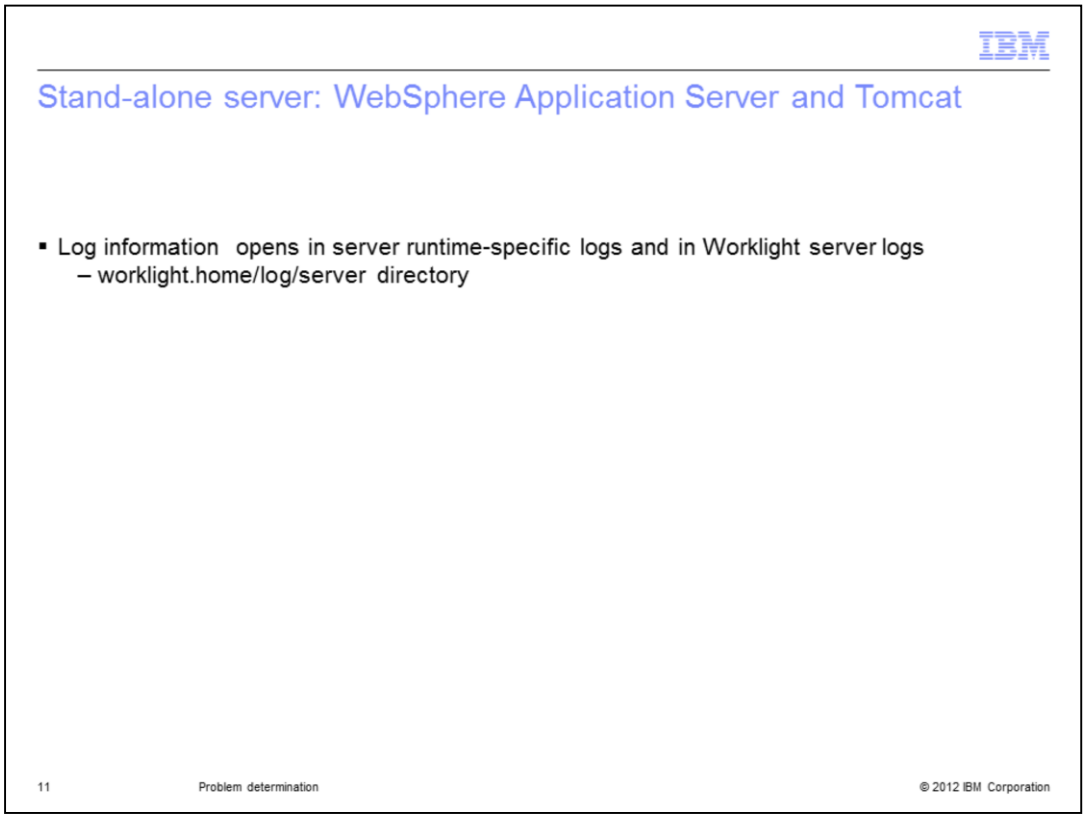


Worklight Studio provides the ability to start and test adapters from within the development environment. This makes it easier to discover and debug issues before they are encountered in an operational environment. When you right-click an Adapter within a Worklight project within Eclipse and select the Run-As action, you are able to select run profiles which can execute a Worklight procedure or a back-end service as required.

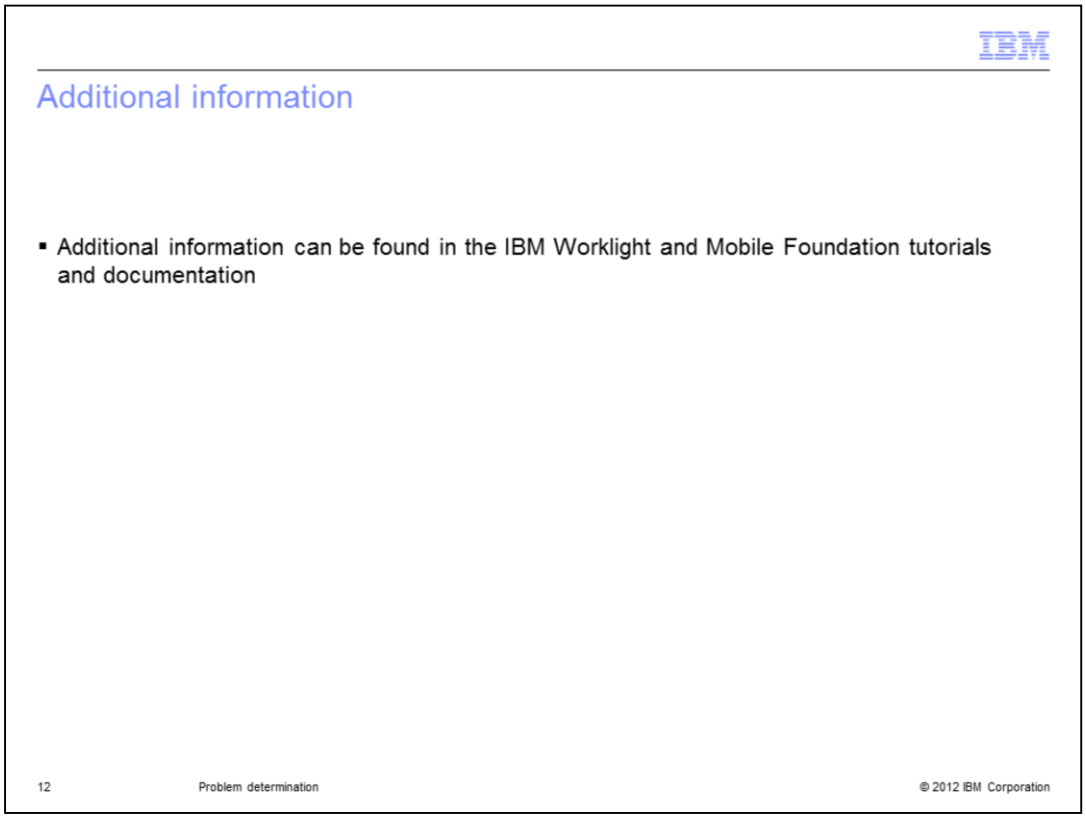
The Worklight API which adapters can use contains a set of procedures which makes it easy to log information to the Worklight server logs.



The following scenarios provide an introduction to problem determination and debugging in the stand-alone Worklight server instances.



When deployed as an application in an enterprise server setting, useful log information can be found in both the enterprise server's log files and in the Worklight server logs which opens under a worklight.home/log/server directory.



Additional information about diagnosing and debugging issues can be found in the IBM Worklight and Mobile Foundation tutorials and documentation.

Trademarks, disclaimer, and copyright information