

Xcode Continuous Integration Guide

Contents

About Continuous Integration in Xcode 4

At a Glance 4

 Install and Set Up the Xcode Service 4

 Connect the Xcode Service to Source Code Repositories 5

 Create and Run Bots 5

 Monitor and Manage Bots 5

Install OS X Server and Configure the Xcode Service 6

Download OS X Server and Xcode 6

Set Up the Xcode Service for Yourself 7

Set Up the Xcode Service for Team Members 9

Enable Access to Your Source Code Repositories 12

Connect to Remote Repositories 12

Host Git Repositories on the Server Running the Xcode Service 13

 Clone a Local Repository from Your Development Mac to the Server 14

 Create a Project and Host its Repository on the Server 15

 Create Git Repositories on the Server and Access Them from Your Development Mac 16

 Push Your Commits to the Hosted Repository on the Server 17

 Use Git to Manage an Unmanaged Workspace Directory on a Development Mac 17

Configure Bots to Perform Continuous Integrations 19

Share Build Schemes 19

Create Bots 20

Follow Best Practices 21

Manage and Monitor Bots from the Log Navigator 22

Manage Bots in the Log Navigator 22

Monitor and Manage Bots in the Bot Viewer 22

Review Integration Details in the Integration Viewer 24

Manage and Monitor Bots from a Web Browser 27

View the Bots Website 28

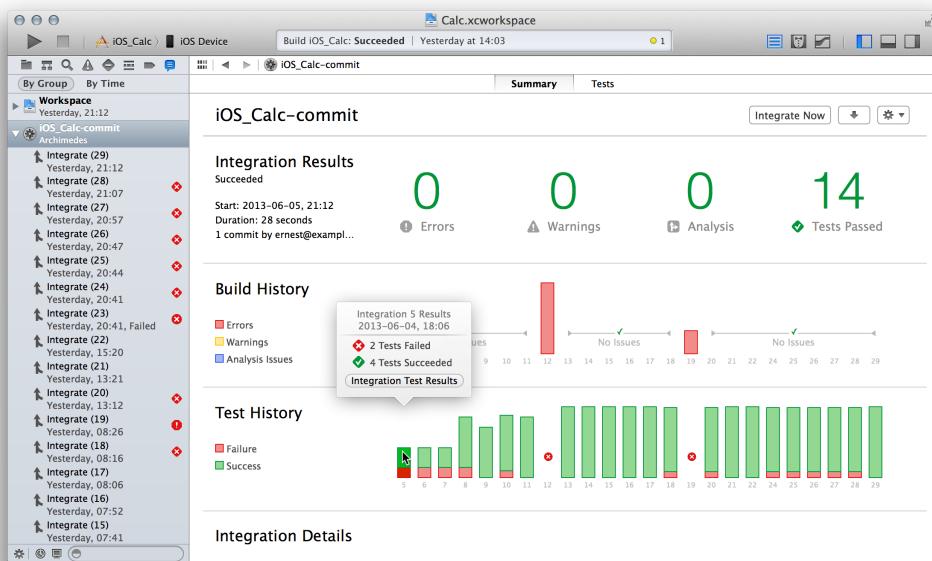
Manage and Monitor Bots from the Bots Homepage 29

[Review Integration Details in a Web Browser](#) 31
[Download Products and Archives from a Web Browser](#) 33
[Watch Your Bots on Big Screen](#) 34

[Document Revision History](#) 36

About Continuous Integration in Xcode

Xcode supports a continuous integration workflow through the Xcode service. The **Xcode service**, available in OS X Server, automates the integration process of building, analyzing, testing, and archiving your app. From your development Mac, you create **bots** that run on a separate server, where they perform these integrations. Bots help ensure that your product is always in a releasable state—and when there's a failure, the service notifies you or the person whose code change caused the failure.



At a Glance

Follow the steps outlined in this document to set up a continuous integration workflow using Xcode and the Xcode service.

Install and Set Up the Xcode Service

Even if you've never set up a server, you'll find the process for setting up the Xcode service in OS X Server to be straightforward.

Relevant chapter: “Install OS X Server and Configure the Xcode Service” (page 6)

Connect the Xcode Service to Source Code Repositories

For bots to perform integrations of projects, the bots must have access to the projects’ source code repositories. The Xcode service supports two popular source control systems: Git and Subversion. You can use Git and Subversion repositories hosted on remote servers, and you can host and use Git repositories on the server with OS X Server installed.

Relevant chapter: “Enable Access to Your Source Code Repositories” (page 12)

Create and Run Bots

Bots are at the center of the automated workflow. Bots build and test products with the schemes of your choosing. With the Xcode service able to access the source code repositories of your projects, you can create and schedule bots to run either periodically or on every source code commit. You can also configure bots to send email notification of the success or failure of their integrations.

Relevant chapter: “Configure Bots to Perform Continuous Integrations” (page 19)

Monitor and Manage Bots

From the log navigator in Xcode on the development Mac, you can manage bots, view their test results, read integration logs, initiate integrations, and download product builds and archives. The Xcode service also hosts a bots website where you and members of your development team can perform these operations.

Relevant chapters: “Manage and Monitor Bots from the Log Navigator” (page 22), “Manage and Monitor Bots from a Web Browser” (page 27)

Install OS X Server and Configure the Xcode Service

To run the Xcode service, install and configure OS X Server and Xcode on a Mac running OS X Mavericks. You can then write code on a separate development Mac and let the server perform continuous integrations of your software products.

The Xcode service advertises itself over Bonjour on your local network. If you and other users will access the Xcode service only from your local network, you can use Bonjour to find the Xcode service. If you need the service to be visible more broadly, ask your DNS server administrator to add records for the new server running OS X Server to a DNS server. After these records are added, users can access the server by its host name, such as `server.mycompany.com`. If your intranet doesn't have a DNS server, you and others can access the server by its local hostname, such as `server.local`.

Note: A continuous integration workflow typically relies on one or more development Mac computers running Xcode and on a separate server running the Xcode service. However, you can install OS X Server and run the Xcode service on your development Mac. Such a configuration can be helpful for evaluating how to adopt continuous integration. Afterward, you'll find it more useful to have a dedicated server running the Xcode service, hosting your repositories, and remotely performing integrations.

Download OS X Server and Xcode

To install OS X Server on a Mac

1. Download OS X Server from the App Store.

Open the App Store app on the Mac that will be your server, and search for OS X Server.

The Mac must run OS X v10.9 or later. If you have an earlier version, click Updates in the App Store toolbar and upgrade OS X before installing OS X Server.

After you download OS X Server from the App Store, the Server app is installed and opens automatically.

- Click Continue in the OS X Server setup window (or click Help to see detailed setup instructions).



- Follow the onscreen instructions to complete the installation.

After you enter the name and password of an administrator account on your Mac, the Server app installs the software it needs, and the Server app configures your Mac as a server.

You must also install Xcode on the server. Even if you don't use Xcode directly on the server while developing software, the Xcode service uses Xcode to run its bots.

To install the Xcode app

- Download Xcode from the App Store. After you download Xcode, it automatically appears in Launchpad.
Your server must run Xcode 5 or later. If you have an earlier version of Xcode, click Updates in the App Store toolbar and upgrade Xcode.

Set Up the Xcode Service for Yourself

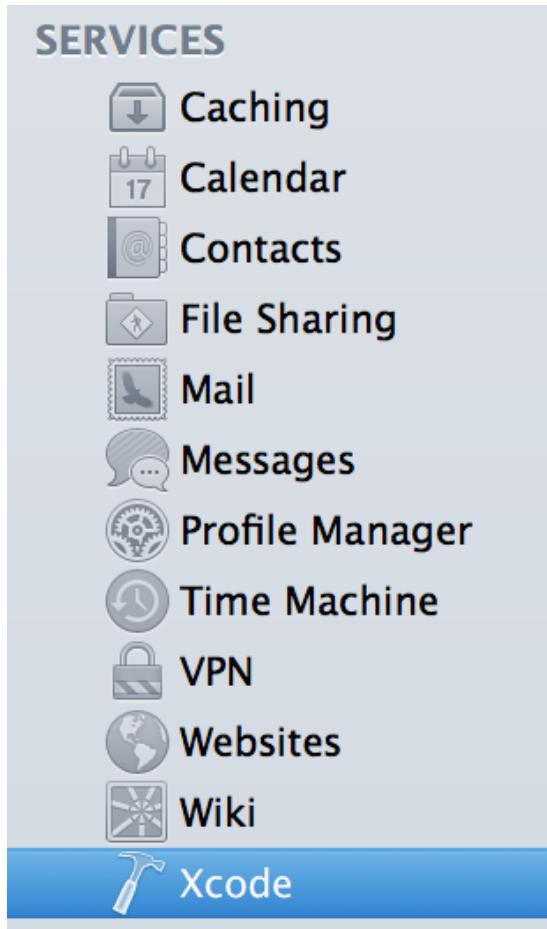
With the Xcode and Server apps installed, you're ready to start the Xcode service.

To start the Xcode service from the Server app

- Launch the Server app (if it's not already running).

You can click the Launchpad icon in the Dock and then click Server.

2. In the Server app sidebar, select the Xcode service.



3. Click Choose Xcode, and select Xcode 5 (or later).

The first time you turn on the Xcode service for a particular server, the service asks you to identify the Xcode release it should use to perform its tasks.

4. Click the On/Off switch to turn on the service.

Add your server to an Apple Developer Program Member Center team to allow the Xcode service to access Member Center assets, such as provisioning profiles and signing certificates to build products for iOS devices. You must be an admin or the agent of the team you want your server to join. (See the chapter “Managing Your Team” in *App Distribution Guide* for information about Apple Developer Program team roles.)

To add a server to an Apple Developer Program Member Center team

1. In the Server app sidebar, select the Xcode service.
2. Click Settings.

3. In the Settings pane, click the Add button, associated with the Developer Teams label.



4. Sign in with your Apple Developer credentials.
5. Choose a team, and click OK.

After adding a team to the Xcode service, you can add provisioned iOS development devices for use when running unit tests.

To add an iOS development device to the Xcode service

1. In the Server app sidebar, select the Xcode service.
2. Connect the device to the server, and wait until it appears in the devices list.

A device must be provisioned to appear in the devices list. (To provision a device, follow the instructions in Provisioning Your iOS Device for Development in *Devices Organizer Help*.)

3. Click the “Add to Team” button next to the device in the list.

On the development Mac, add an account for the server. With the server account added to your development Mac, you can check the status of bots in the Xcode log navigator. You can also create new projects and host them in Git repositories on the server.

To add an OS X Server account to Xcode on a development Mac

1. Choose Xcode > Preferences on your development Mac.
2. Click Accounts.
3. Click the Add button (+), and choose Add Server.
4. Select the server from the server list or enter a server address, and click the Next button.
5. Specify your connection credentials to the server, and click Add.

If the server is successfully enabled, you can click the server address link in the Accounts preferences in Xcode, and Safari will launch and show the bots website hosted by the server.

Set Up the Xcode Service for Team Members

If you are part of a team of developers, you can give the members user accounts on the server running OS X Server. They can then create and manage bots, share and manage source code repositories, and use other services in OS X Server.

To add a user account to the server

1. In the Server app sidebar, select Users (under Accounts).

For more details about creating user accounts, click the Help button in the lower-right corner of the Users pane.

2. Click the Add button (+).
3. Enter the user's account information in the New User pane.
4. Click Create.

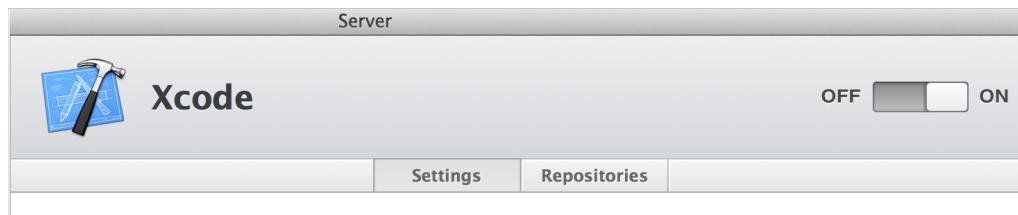
You can also add and manage groups of user accounts by selecting Groups (under Accounts) in the Server app sidebar. For more details, click the Help button in the lower-right corner of the Groups pane.

In Xcode Accounts preferences on their development Mac computers, team members can add their account credentials for the server.

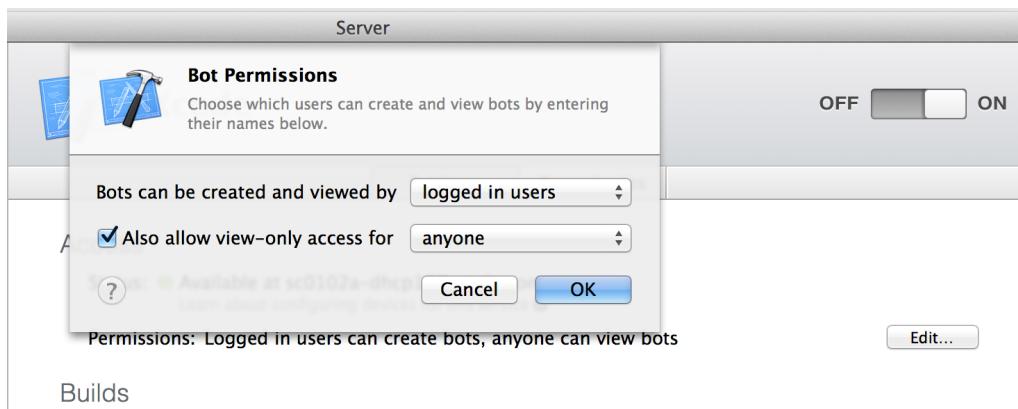
Anyone running Xcode who has access to the server can potentially create and view bots. You can restrict which users have access to bots by clicking the Edit button in the Settings pane in Xcode service.

To change permissions for who can create or view bots

1. In the Server app sidebar, select the Xcode service.
2. Click Settings.



3. Click the Edit button for Permissions.
4. Choose which users can create and view bots.



If you choose “anyone” from the upper pop-up menu, guests and every authenticated user can view bots, create, edit, and delete bots, and download items. “Logged in users” includes local users and directory users, all of whom need to authenticate to have access to bots. Clicking “only some users” allows you to specify existing users or groups. If you don’t know the exact name of a user or group, clicking Add (+) and entering a space allows you to browse and filter the directory.

5. If you chose to restrict bot creation, you can choose to restrict view-only access to bots.

Users with view-only access can view the website, but they can’t create bots, manage bots, or initiate integrations. People who particularly benefit from having view-only access to bot activity are software testers, project managers, and seed coordinators.

6. Click OK.

As described in [“Manage and Monitor Bots from a Web Browser”](#) (page 27), these settings also affect users of the bots website hosted by the server.

Enable Access to Your Source Code Repositories

The Xcode service operates on projects contained in source code repositories. The Xcode service supports two popular source control systems: Git and Subversion. You can use Git and Subversion repositories hosted on remote servers, and you can host and use Git repositories on the server running the Xcode service.

Connect to Remote Repositories

If you have projects in Git or Subversion repositories on remote servers, you can store your credentials for them in Accounts preferences in Xcode on your development Mac. Then you don't have to reenter your credentials every time you need to access the repositories.

To add a remote repository's credentials to a development Mac

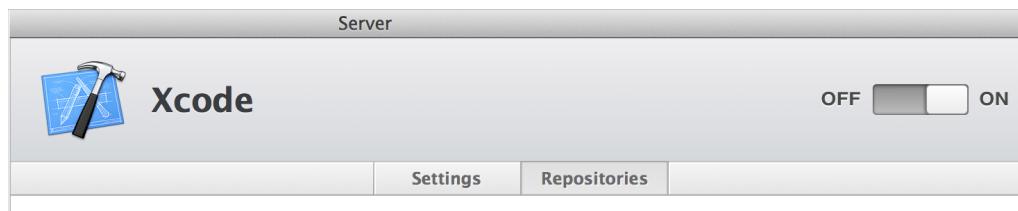
1. Choose Xcode > Preferences on your development Mac.
2. Click Accounts.
3. Click the Add button (+), and choose Add Repository.
4. In the text field, enter the URL for the repository (for example, `svn+ssh://svn.example.com/ProjectName` or `https://example.com/git/repository.git`), and click the Next button.
5. Choose either Git or Subversion from the pop-up menu.
6. Enter your user name and password in the Repository pane of Accounts preferences, and click Add.

For the Xcode service to perform integrations on your projects, it must also have access to their repositories. Configure the Xcode service to connect to your remote repositories.

To add a remote repository to the Xcode service

1. In the Server app sidebar, select the Xcode service.

2. Click Repositories.



3. Click the Add button (+).
4. Select either "Connect to a Git Repository" or "Connect to a Subversion Repository."
5. Enter a name for the repository.

Use any name that helps you identify the repository. The name will appear in the Remote Repositories list in the Xcode service.

6. Enter the URL for the repository (for example, `svn+ssh://svn.example.com/ProjectName` or `https://example.com/git/repository.git`).
7. Select the type of authentication, and enter your authentication credentials.

Click None if the repository requires no authentication. Click Password if the repository requires a user name and password. Enter them after selecting this method. Click SSH Key if the repository requires SSH key-based authentication and encryption. Enter the user name, and copy the public key to the server.

8. Click Create.

The name and address of the repository appear in the Remote Repositories list.

Host Git Repositories on the Server Running the Xcode Service

You can create and share Git repositories on the computer with OS X Server installed. Git is a free, open-source version management system for code. (For more information about Git, see the [project web site](#).) If you work on a team of developers, they can share and manage their code changes in these repositories.

Configure the Xcode service to allow users to create and gain access to hosted source code repositories.

To configure repository access in the Xcode service

1. In the Server app sidebar, select the Xcode service.
2. Click Repositories.
3. Make sure the checkbox for "Allow access to hosted repositories" is selected. Click the associated Edit button to select which protocols can be used to access the hosted repositories.

By default, the HTTP and HTTPS protocols are selected.

You can also select SSH. If you select SSH, the Xcode service displays a dialog asking whether to allow remote login using SSH. Click Allow.

HTTP and HTTPS offer the most flexibility, because these protocols work in most network environments. Your authentication credentials are encrypted with HTTP, but the remainder of your transactions—consisting mainly of your source code—aren’t encrypted. With HTTPS, your transactions are encrypted along with your credentials, but you need a valid certificate signed by a public certificate authority.

SSH encrypts credentials and transactions. However, SSH isn’t as readily available as HTTP and HTTPS across network environments.

SSH is a good, secure choice and is useful if your organization uses SSH keys for authentication. HTTPS is a good, secure choice if you have SSL certificates for the server. HTTP is a good choice if you don’t have SSL certificates, and aren’t planning on accessing code outside your local network or over an encrypted connection. For more information about these protocols, see [OS X Server: Advanced Administration](#).

4. Click the lower Edit button to select the users who can create hosted repositories.
5. In Accounts preferences in Xcode on your development Mac, add your account credentials for the server (if you haven’t already done so).

Note: If you allow HTTPS access, anyone using HTTPS to access a hosted repository is presented with a certificate dialog in Xcode. To access the repository from your development Mac, click Show Certificate in the certificate dialog. Select the Always Trust option, and click Continue.

If you work with a team of developers, you can give them accounts for accessing repositories on the server (as described in [“Set Up the Xcode Service for Team Members”](#) (page 9)).

Clone a Local Repository from Your Development Mac to the Server

If you use a Git repository local to your development Mac, clone the repository to the server running the Xcode service, thereby allowing the Xcode service to operate on the repository.

To clone a local repository to a server running the Xcode service

1. Open the project on your development Mac, and choose Source Control > *ProjectName* > Configure *ProjectName*.
2. Click Remotes.
3. Click the Add button (+).
4. Choose Create New Remote.
5. Select the server.

6. Enter a name for the remote repository.

Use any name that helps you identify the repository. The name will appear in the Remotes list on your development Mac when you need to select this repository. For example, when you choose Source Control > Push and Source Control > Commit, the name appears in a pop-up menu that allows you to specify the remote repository.

7. Click Create.

The cloned repository appears in the Hosted Repositories list in the Xcode service.

8. Click Done.

Your local repository is copied to the server.

If you work with a team of developers, you can give them accounts on the server to share your repository, as described in [“Set Up the Xcode Service for Team Members”](#) (page 9).

Create a Project and Host its Repository on the Server

When you create a project on your development Mac, you can simultaneously create a repository for it on the server.

To create a project with a Git repository on a server running the Xcode service

1. In the Accounts preferences in Xcode on your development Mac, add your account credentials for the server (if you haven’t already done so).

By default, the Xcode service allows hosted repositories to be created by logged in users. Adding your account credentials for the server in your Accounts preferences in Xcode allows you to log in automatically when creating a Git repository on the server.

2. Choose File > New > Project.

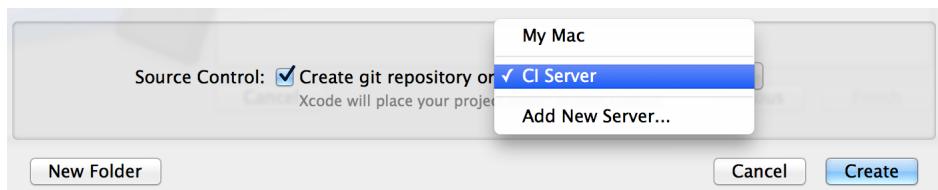
3. Choose a template for your project, and click Next.

4. Specify options for the project, and click Next.

5. Specify the location for the local working copy of the project.

6. For the Source Control option, select “Create git repository on.”

7. Use the associated pop-up menu to choose the server on which to host the repository.



If the server doesn't appear in the list, look—or ask the server administrator to look—in the Repositories pane of the Xcode service to see whether you have permission to create a repository. Depending on how the Xcode service is configured, hosted repositories can be created by logged in users, anyone, or only specified users.

8. Click Create.

On the server, the name of the project appears in the Hosted Repositories list of the Xcode service. If you work with a team of developers, you can give them accounts to share your repository (as described in [“Set Up the Xcode Service for Team Members”](#) (page 9)).

Create Git Repositories on the Server and Access Them from Your Development Mac

You can create sharable Git repositories on the server running the Xcode service.

To host a new Git repository on a server running the Xcode service

1. In the Server app sidebar, select the Xcode service.
2. Click Repositories.
3. Click the Add button (+).
4. Choose “Host a Git Repository.”
5. Enter a name for the repository.

Use any name that helps you identify the repository. This name will appear in the Hosted Repositories list in the Xcode service and will be part of the access URL.

6. Identify the users allowed to access the repository.

Select “All authenticated users” to give access to all users who can log in to the local server or the directory.

Select “Only some users” to set up an access list for users on the local server or in the directory. You can choose who can read or who can read and write to the repository.

7. Click Create.

The new repository appears in the Hosted Repositories list. If you want to change user access permissions later, select the repository from the Hosted Repositories list and click the Edit button.

You can add your credentials for this repository to your development Mac by using Accounts preferences in Xcode on that Mac. If you work with a team of developers, you can give them accounts on the server to share the repository, as described in [“Set Up the Xcode Service for Team Members”](#) (page 9).

If you have an existing Git repository on your local development Mac, you can clone it to the new repository on the server.

To add a Git repository hosted by the Xcode service to a project on a development Mac

1. Open the project on your development Mac, and choose Source Control > *ProjectName* > Configure *ProjectName*.
2. Click Remotes.
3. Click the Add button (+).
4. Choose Add Remote.
5. Select the server.
6. Enter the name and address for the remote repository.
7. Click Add Remote.
8. Click Done.

Push Your Commits to the Hosted Repository on the Server

After you configure your development Mac to use a Git repository on the server, a commit operation adds your changes to your local repository. As with any remote Git repository, you must also perform a push operation to add your committed changes to the repository on the server. For example, when you choose Source Control > Commit on your development Mac, select the “Push to remote” option, specify the remote repository in the pop-up menu, and click Commit Files.

Use Git to Manage an Unmanaged Workspace Directory on a Development Mac

When you create an Xcode workspace or project, you have the option of including a Git repository in the generated workspace directory. If you didn’t select that option, your workspace directory doesn’t include a Git repository. To share a workspace directory with others using the Git version control system, you must initialize a Git repository in that directory.

To initialize a Git repository in a workspace directory

1. In the Terminal app, run the `git init` command inside the workspace directory:

```
cd <workspace_directory_path>
git init
```

2. Identify the files you want to track in the repository with the `git add` command:

```
git add <files>
```

3. Add the files to the repository with the `git commit` command:

```
git commit -m "<workspace_directory> initial commit"
```

For example, these commands initialize a Git repository inside the Sketch workspace directory, identify the files in the directory to be tracked (excluding some), and add the files to the newly created repository:

```
hedy: Desktop $ cd Sketch
hedy: Sketch $ git init
Initialized empty Git repository in /Users/ernest/Desktop/Sketch/.git/
hedy: Sketch $ cat > .gitignore
.DS_Store
xcuserdata
^D          # Control-D
hedy: Sketch $ git add .
hedy: Sketch $ git commit -m "Sketch initial commit"
[master (root-commit) db941e7] Sketch initial commit
 73 files changed, 13157 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Arrow.tiff
 create mode 100644 Circle.tiff
...
```

To allow the Xcode service to operate on the repository, clone the repository to your server, as described in ["Clone a Local Repository from Your Development Mac to the Server" \(page 14\)](#).

Configure Bots to Perform Continuous Integrations

Bots are processes run by the Xcode service to perform integrations on the current version of a project in a repository. Integrations consist of building, analyzing, testing, and archiving the apps (or other software products) defined in your projects. With the Xcode service able to access the source code repositories of those projects, you can configure bots to perform continuous integrations on them.

On a development Mac, a scheme defines what operations are performed by an integration. To automate an integration, you share its scheme and you create a bot to perform scheduled integrations. A bot can automatically perform integrations:

- Every time a change is published to the repository
- On a regular schedule

Share Build Schemes

A scheme specifies which targets to build for a project, which build configuration to use, and which executable environment to use when the product is launched. When you create a new iOS or OS X project, Xcode creates a default scheme that includes settings to perform these actions:

- Analyze, which performs static code analysis
- Test, which runs the unit test cases that you implement
- Archive, which builds an archive of the product that the scheme built

For the Xcode service to perform these actions, you must share their scheme. A **shared scheme** is one that you publish in a repository, along with the other shared project files.

To share a scheme

1. On your development Mac, check out and open the project that contains the scheme to share.
2. Choose Product > Scheme > Manage Schemes.
3. Select the Shared option for the scheme to share, and click OK.
4. Choose Source Control > Commit.
5. Select the Shared Data folder.

6. Enter your commit message in the text field.
7. Select the “Push to remote” option (if your project is managed with Git).
8. Click the Commit Files button.

For more information about managing schemes, see *Scheme Configuration Help*.

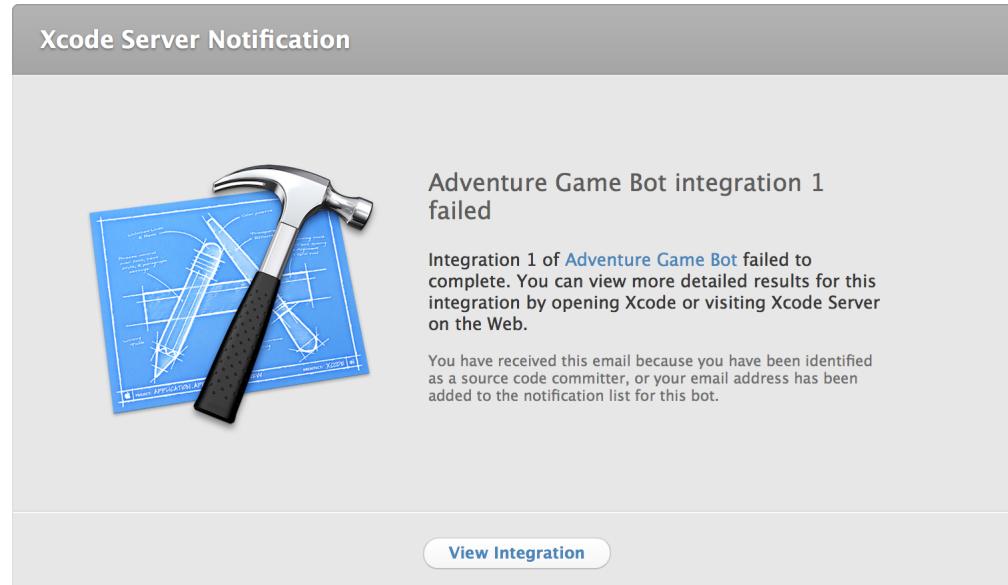
Create Bots

After sharing a scheme, create a bot to perform the integrations.

To create a bot

1. On your development Mac, open the project containing the scheme that defines the actions to automate.
2. Choose Product > Create Bot, and specify the identifying attributes of the bot.
Leave the “Integrate immediately” option enabled to see the results of your first integration in the log navigator.
3. Click Next.
If you have not already added your repository to Xcode in Accounts preferences, enter your authentication credentials as prompted.
4. Specify the integration schedule, the actions the bot is to perform, and whether to clean products before building.
You can schedule the bot to perform its integrations periodically (hourly, daily, or weekly), on every commit, or manually. If you enable the Cleaning option, the bot won’t reuse the previous build.
5. Click Next.
6. For an iOS app, choose what kinds of devices or simulators the bot will test on.
The devices must be connected to the server for the test action to complete.
7. Click Next.

8. Specify who should receive email notifications from the bot (such as the notification shown in the screenshot).



9. Click Create Bot.

As explained in the following chapter, you can use the log navigator to manually start the bot, edit the bot, and delete it. As explained in ["Manage and Monitor Bots from a Web Browser"](#) (page 27), you can also use a web browser to manually start the bot, edit the bot, and delete it.

Follow Best Practices

To take advantage of continuous integration in your product development workflow, follow these practices:

- **Develop unit-test suites and test cases.** After developing unit tests, include them in schemes for your bots to run. To help ensure that the changes you make aren't broken by you or others later, complement those changes with unit tests that determine whether a method or a set of methods used in a sequence functions as intended.
- **Perform static analysis.** Include static analysis in your integrations. Static analysis is a deep examination of your code, following code paths that your app may not follow during normal development. This process uncovers hard-to-find coding errors and also identifies areas in your code that don't follow recommended API usage, such as Foundation and AppKit idioms.
- **Ensure that your product builds and is packaged correctly.** Archive your product after making major changes, especially structural changes, such as adding or removing files. Let your bots archive for you automatically. The ability to build and archive your product is a main indicator of the correctness of your code changes.

Manage and Monitor Bots from the Log Navigator

The log navigator provides access to information about your bots and the integrations they've performed. You can edit, delete, and create bots from the log navigator, and you can initiate and cancel any of their integrations. By selecting bots and their integrations in the log viewer, you can view information about them in the editor area of the Xcode workspace, where you can also edit and delete a bot, initiate an integration, and download its archives.

Manage Bots in the Log Navigator

In Xcode on your development Mac, choose View > Navigators > Show Log Navigator. Click By Group to see status information gathered under each of the bots. Control-click a bot to display a shortcut menu that allows you to:

- Edit the bot
You can change its scheme, name, schedule, actions, email recipients, and test devices.
- Delete the bot
When you remove a bot, you stop its future integration actions and remove existing builds and archives.
- Start an additional integration immediately
- Cancel an integration in progress

You can also create a bot from the shortcut menu in the log navigator.

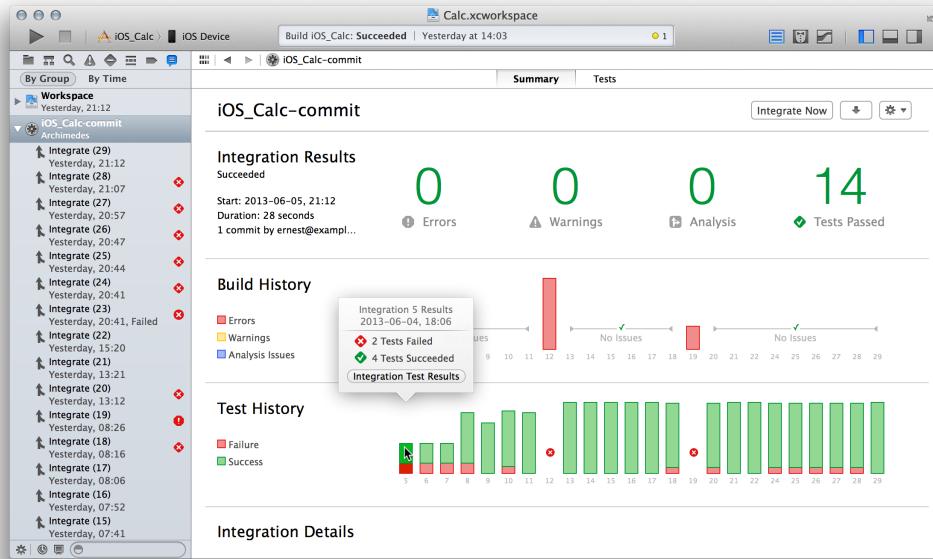
Monitor and Manage Bots in the Bot Viewer

You can view summaries of a bot's integrations by selecting it in the log navigator.

To view a summary of integration results for a bot

1. Click By Group at the top of the log navigator.
Status information is gathered for each of the bots.
2. Select the bot whose integrations you want to view.

In the editor area of the workspace window, the bot viewer displays a summary of the integrations it has performed along with any build errors and warnings, static analyzer problems, and unit test failures.



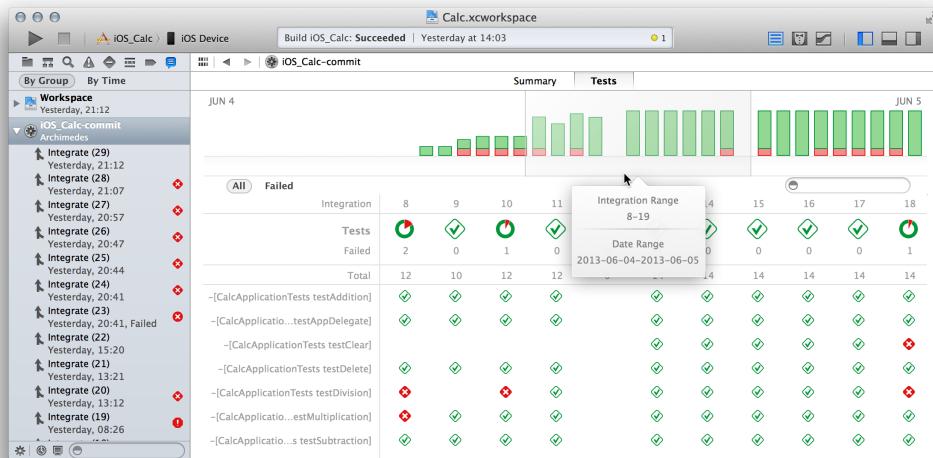
You can perform these operations in the bot viewer:

- To start an integration: Click the Integrate Now button.
- To download the bot's last archive: Click the down-pointing arrow button.
- To edit or delete a bot or to view the bot's summary webpage: Use the Action pop-up menu (with the gear).

In the editor area, the Summary pane of the bot viewer displays these elements:

- **Integration Results:** The results of the bot's last integration, including the number of errors, warnings, static analysis issues, and failed tests.
- **Build History:** A bar graph depicting the errors, warnings, and static analysis issues the bot encountered when building. Click a bar to get a tally of the issues.
- **Test History:** The number of successful and failed test cases the bot has performed. Click a bar to get a tally of the passed and failed tests.
- **Integration Details:** The errors, warnings, and analysis issues of the last integration.

The Tests pane of the bot viewer provides a tabulated list of the unit test results.



The Tests pane displays these elements:

- Test history chart:** Located at the top of the integration viewer, this chart depicts results for the integrations the bot has performed. Drag the highlighted area to select the range of test runs to display in the test results summary section.
- Test results summary:** Below the test history chart, the test results summary tabulates the test results of the integrations highlighted in the history chart. Click the Failed button to list only the tests that failed. Click the All button to list all the tests.

Review Integration Details in the Integration Viewer

You can view details about an integration by selecting it in the log navigator.

To view details about an integration

- Click By Group at the top of the log navigator.
Status information is collapsed under each of the bots.
- Select the bot whose integrations you want to view.
- Click the disclosure triangle to the left of the bot to display its integrations.

You can quickly check the status of an integration. To the right of each integration listed in the log navigator, Xcode displays an icon that indicates whether there was an error, a warning, a static analysis issue, or an integration failure.

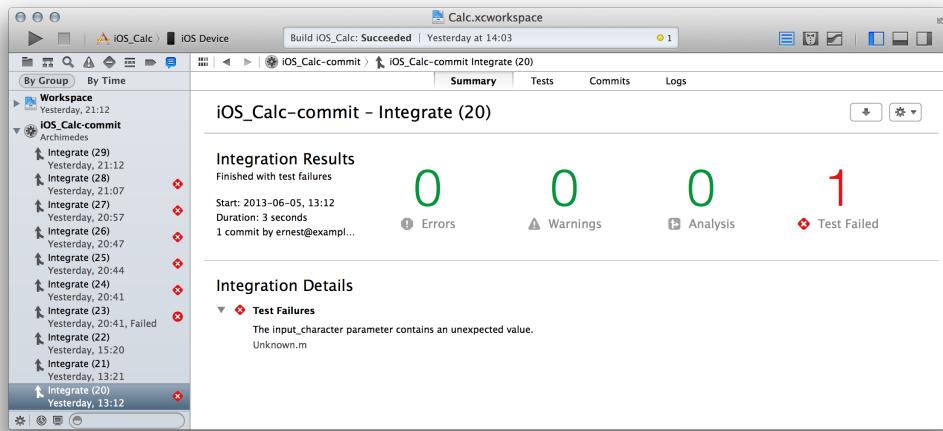
Manage and Monitor Bots from the Log Navigator

Review Integration Details in the Integration Viewer

4. Select a specific integration.

In the editor area of the workspace window, the integration viewer displays a summary of the integration's results.

The number of errors, warnings, static analysis issues, and test-case failures for that integration are summarized in the Integration Results section. The Integration Details section provides details about each of the integration issues. If the integration contains errors, warnings, static analysis issues, or test failures, click Tests or Logs to view their details.



5. Click Tests to view a list of unit tests and their pass or fail status.

If you have multiple test devices, you can compare their results across columns.

Test Results				Device Summary			
	Device	OS	Processor	Tests Total	Tests Passed	Tests Failed	4 Devices Tested
Model	QBear_iPod_touch	6.1.3 / armv7	Ernest_iPad2	2	2	0	1/2 iPads Failed
Tests	7		7	7	7	0	1/2 iPhones Failed
Failed	2		2	2	2	0	
-[CalcApplic...testAddition]	✓		✓	✓	✓	✓	
-[CalcApplic...ppDelegate]	✓		✓	✓	✓	✓	
-[CalcApplic...sts testClear]	✗		✗	✗	✗	✗	
-[CalcApplic...ts testDelete]	✓		✓	✓	✓	✓	
-[CalcApplic...testDivision]	✗		✗	✗	✗	✗	
-[CalcApplic...ultiplication]	✓		✓	✓	✓	✓	
-[CalcApplic...Subtraction]	✓		✓	✓	✓	✓	

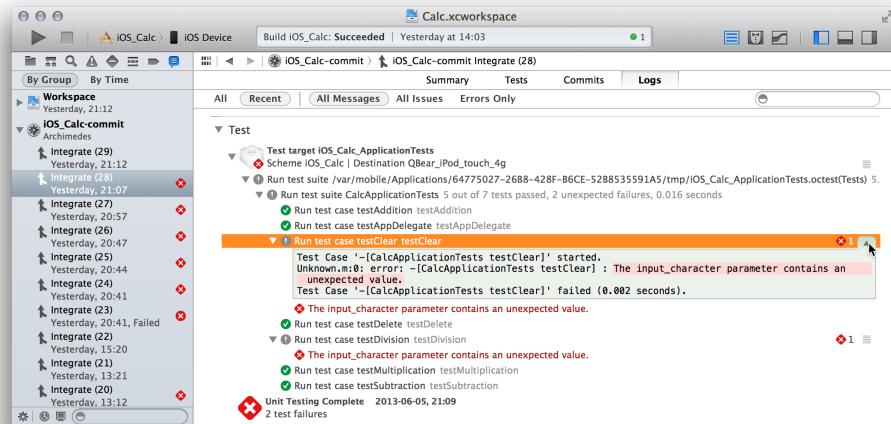
6. Click Commits to view details about the new commits included in the integration.

By clicking the "Show modified files" button, you can view the files that are part of the commit.

Manage and Monitor Bots from the Log Navigator

Review Integration Details in the Integration Viewer

7. Click Logs to view the logs of the actions that are part of the integration.

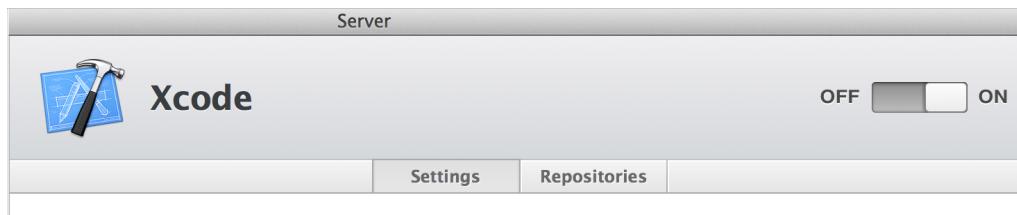


Manage and Monitor Bots from a Web Browser

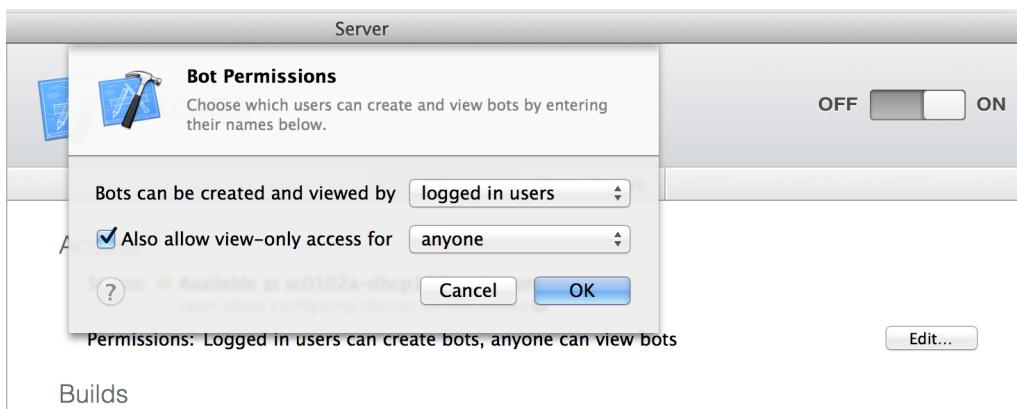
The bots website hosted on your server provides summaries of its bots' activities along with details about their integrations. From the bots website, you and your team can edit, delete, and create bots; initiate any of their integrations; and download archives and product builds. (Products contain just the app, and archives contain the Xcode project.)

To specify who can view and use the bots website

1. In the Server app sidebar, select the Xcode service.
2. Click Settings.



3. Click the Edit button for Permissions.
4. Use the upper pop-up menu to specify who can create and manage bots as well as download their archives and products.



If you choose “anyone” from the upper pop-up menu, any visitor to the site can view bots, create, edit and delete bots, and download items.

If you choose “logged in users” or “only some users” from the upper pop-up menu, an unauthenticated



user sees a Bots homepage devoid of data. The lock icon () in the top-right corner of the webpage allows the user to log in with a valid user name and password. (See [“Set Up the Xcode Service for Team Members”](#) (page 9) for information about creating user accounts on the server.)

If you choose “only some users” from the upper pop-up menu, you are presented with a table of users and groups. Add and remove users and groups to suit your needs.

5. Use the checkbox and the lower pop-up menu to allow or restrict view-only access to the website.

Users with view-only access can view the website, but they can’t create bots, manage bots, or initiate integrations. People who particularly benefit from having view-only access to bot activity are software testers, project managers, and seed coordinators.

View the Bots Website

To view the bots website, use the address `hostname/xcode/bots`, where `hostname` is the Internet domain name for the server (such as `server.mycompany.com`) or its local hostname (such as `server.local`).

The Bots homepage displays the status of all of the bots on the server. The page shows a summary of the last integration, the amount of time before the next integration begins, links to the downloads from the last integration, and a list of the bots running on the server.

The screenshot shows the Xcode Bots homepage with the following details:

- Latest Integration:** iOS_Calc-hourly, Integration 8. Status: Failed. Errors: 0, Warnings: 0, Analysis: 0.
- Next Integration:** iOS_Calc-hourly, Integration 9. Scheduled for 00:00:40.
- Latest Downloads:**
 - iOS_Calc-commit (Integration 29): Product (Created 6/5/2013), Archive (Created 6/5/2013).
 - Archive (Created 6/5/2013): 47 KB.
 - Product (Created 6/5/2013): 153 KB.
- Bots List:**

Name	Status	Next	Last
iOS_Calc-commit	Succeeded	Yesterday at 9:15 PM	Integrate
iOS_Calc-hourly	Succeeded	Today at 2:45 PM	Integrate
Mac_Calc-commit	Finished with warnings	Yesterday at 9:12 PM	Integrate

From this page you can further select information to view by clicking:

- **View Summary** to see the summary page of the latest integration
- **View Archives** to see a complete list of archives and products from past integrations
- to view the big screen (described in [“Watch Your Bots on Big Screen”](#) (page 34))

-  to view help for the page
- **All** to see all of the bots in the table below
- **Failed** to see only the bots with failed tests
- **Passed** to see only the bots with passed tests
- Any **name** in the table to view a summary of that bot's integrations

You can perform tasks on the page by clicking:

- **Integrate Now** to start an additional integration
- **Product** to download the latest build
- **Archive** to download the latest archive
-  to create a new bot (as described in onscreen instructions)
-  to log in to or out of the website
- **Integrate** in a bot's row to initiate an additional integration by the bot

You can also access the bots website from the log navigator on your development Mac.

To access the bots website from the log navigator

1. In Xcode on your development Mac, choose View > Navigators > Show Log Navigator.
2. Click By Group at the top of the log navigator.
All integrations for a bot are grouped under the bot name.
3. Control-click a bot in the log navigator, and choose "View Bot in Browser" from the shortcut menu.
A web browser opens to the status page for the selected bot.
4. Click Bots in the navigation bar at the top of the webpage to see the bots homepage.

Manage and Monitor Bots from the Bots Homepage

You can view summaries of a bot's integrations by selecting the bot name on the bots homepage.

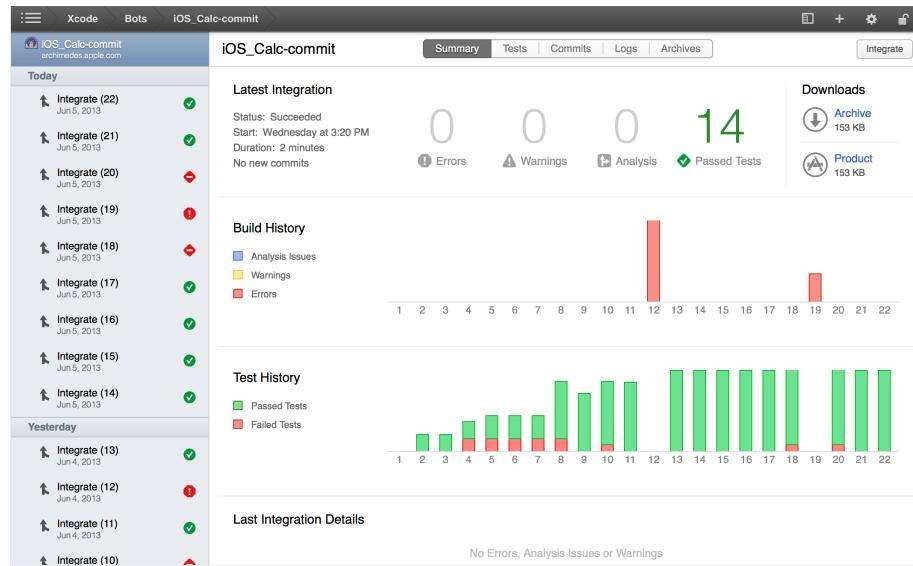
To view a summary of integration results for a bot in a web browser

1. Display the bots website (hostname/xcode/bots).

Hostname is the Internet domain name for the server (such as `server.mycompany.com`) or its local hostname (such as `server.local`).

- Click the name of any bot in the table.

The browser window displays a summary of the integrations performed by the bot along with any build errors and warnings, static analyzer problems, and unit test failures. The results of the most recent integration appear near the top of the window.



The Summary pane displays these elements:

- Latest Integration:** The results of the bot's last integration, including the number of errors, warnings, static analysis issues, and failed tests
- Downloads:** Links for the latest integration's archive and product
- Build History:** A bar graph depicting the errors, warnings, and static analysis issues the bot encountered when building; click a bar to get a tally of the issues
- Test History:** The number of successful and failed test cases the bot has performed; click a bar to get a tally of the passed and failed tests
- Last Integration Details:** The errors, warnings, and analysis issues of the last integration

From this page you can click:

- Tests** to view a complete list of the bot's tests and the pass-or-fail status of the tests
- Commits** to see a list of all of the commits for the integration
- Logs** to view the information logged for all of the bot's activities

- **Archives** to access past archives and product builds from the bot
-  to view the big screen (described in “[Watch Your Bots on Big Screen](#)” (page 34))
-  to create a new bot (as described in onscreen instructions)
-  to delete the bot, to edit its settings, and to view help for the page
-  to log in to or out of the website

You can also access the webpages for a bot from the log navigator on your development Mac.

To access the webpages for a bot from the log navigator

1. In Xcode on your development Mac, choose View > Navigators > Show Log Navigator.
2. Click By Group at the top of the log navigator.
All integrations for a bot are grouped under the bot name.
3. Control-click a bot in the log navigator, and choose “View Bot in Browser” from the shortcut menu.
A web browser opens to the summary page for the selected bot.

Review Integration Details in a Web Browser

You can view details about an integration by selecting it from a bot webpage.

To view details about an integration in a web browser

1. Display the bots website (`hostname/xcode/bots`).
2. Click the name of any bot in the table.

Your browser window displays a summary of the integrations performed by the bot. The results of the most recent integration appear near the top of the window.

3. Select any integration listed in the bot page sidebar.

Manage and Monitor Bots from a Web Browser

Review Integration Details in a Web Browser

When you select an integration, a Summary pane shows the status of the integration, any integration issues, and links to download the integration's archive and build. If there are errors, warnings, static analysis issues, or test failures, use the Tests and Logs panes to find out more about them.

The screenshot shows the 'Summary' tab for the 'iOS_Calc-commit' integration. It lists several recent integrations, each with a green checkmark indicating success. The 'Latest Integration' section shows a status of 'Finished with build errors' at 'Tuesday at 7:56 PM' with a duration of '2 minutes' and '1 commit'. Below this, the 'Last Integration Details' section shows an error log with three entries: 'Property 'displayField' not found on object of type 'CalcViewController'' repeated twice and followed by 'Property 'displayField' not found on object of type 'CalcViewController''. At the bottom right, there are buttons for 'Archive' (153 KB) and 'Product' (153 KB).

- Click Tests to view the list of the integration's tests and their pass or fail status.

If you have multiple test devices, you can compare their results across columns.

The screenshot shows the 'Tests' tab for the 'iOS_Calc-commit' integration. It displays the following statistics: 14 Total Tests, 0 Failed Tests, and 14 Passed Tests. Below this, it shows device details: 'iPad 2 Wi-Fi + 3G (CDMA)' and 'iPod touch (4th generation)'. A table then lists individual test results categorized under 'Family' and 'Processor'. The table includes columns for 'Family', 'Processor', 'Model', 'Tests', 'Failed', and 'Passed'. Each row shows a specific test name and its outcome across the two devices.

- Click Commits to view the new commits that integration includes.
- Click Logs to view the logs of the actions that are part of the integration.
- Click Archives to view a list of the products and archives that the bot has generated.

The screenshot shows the 'Archives' tab for the 'iOS_Calc-commit' integration. It lists the following items:

- Type: Product, Integration: 29, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 29, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 28, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 28, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 27, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 27, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 26, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 26, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 25, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 25, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 24, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 24, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB
- Type: Product, Integration: 23, Date: 6/5/2013, Name: iOS_Calc-commit.ipa, Size: 47 KB
- Type: Archive, Integration: 23, Date: 6/5/2013, Name: Archive.xcarchive.zip, Size: 153 KB

8. Click the Product or Archive link to download the item, or click the Trash icon to delete the integration's archives.

In the top-right corner of this page you can click:

-  to view the big screen (described in “[Watch Your Bots on Big Screen](#)” (page 34))
-  to create a new bot (as described in onscreen instructions)
-  to delete the bot, to edit its settings, and to view help for the page
-  to log in to or out of the website

You can also access the integration details webpage from the log navigator on your development Mac.

To access the integration webpages from the log navigator

1. In Xcode on your development Mac, choose View > Navigators > Show Log Navigator.
2. Click By Group at the top of the log navigator.
All integrations for a bot are grouped under the bot name.
3. If necessary, click the disclosure triangle for a bot to display a list of all of its integrations.
4. Control-click an integration in the log navigator, and choose “View Integration in Browser” from the shortcut menu.
A web browser opens to the status page for the selected integration.

Download Products and Archives from a Web Browser

The bots website facilitates the distribution of product builds and archives to testers and other team members. Products contain just the app, and archives contain the Xcode project.

To download a bot's recent archives from a web browser

1. Display the bots website (`hostname/xcode/bots`).
2. Click the name of any bot in the list.
Your browser window displays a summary of the integrations performed by the bot.
3. Click Archives to display the bot's Archives pane.

- Click a download link for any integration.

You can flag the archives of a specific integration to identify them, for example, as release candidates. What the flag means is up to you and your organization.

	Type	Integration	Date	Name	Size	
Integrate (22)	Product	22	6/5/2013	iOS_Calc-commit.ipa	47 KB	
Integrate (21)	Archive		6/5/2013	Archive.xcarchive.zip	153 KB	
Integrate (20)	Product	21	6/5/2013	iOS_Calc-commit.ipa	47 KB	
Integrate (19)	Archive		6/5/2013	Archive.xcarchive.zip	153 KB	
Integrate (18)	Product	20	6/5/2013	iOS_Calc-commit.ipa	47 KB	
Integrate (17)	Archive		6/5/2013	Archive.xcarchive.zip	153 KB	
Integrate (16)	Product	18	6/5/2013	iOS_Calc-commit.ipa	47 KB	
Integrate (15)	Archive		6/5/2013	Archive.xcarchive.zip	153 KB	
Integrate (14)	Product	17	6/5/2013	iOS_Calc-commit.ipa	47 KB	
	Archive		6/5/2013	Archive.xcarchive.zip	153 KB	

Watch Your Bots on Big Screen

To keep apprised of your bots' status on a large or dedicated display, view them on big screen.

To view bots on big screen

Do one of the following:

- Display the big screen website (`hostname/xcode/bigscreen`).
Hostname is the Internet domain name for your server (such as `server.mycompany.com`) or its local hostname (such as `server.local`).
- From the bots website, click the leftmost button in the navigation bar, and choose Big Screen.
- Click the big screen icon () in the top-right corner of any window in the bots website.

The webpage can be made full screen, and it can be displayed using AirPlay mirroring. Big screen cycles among all of the server's bots, displaying the most pertinent status information for each bot.

The screenshot displays a web-based interface for managing and monitoring bots. On the left, a sidebar lists several bots with their status icons and names:

- Sybil Commit Bot (green checkmark)
- Sybil Hourly Bot (green checkmark)
- Dragonfly Commit Bot (yellow warning)
- Dragonfly Hourly Bot (yellow warning)
- Dragonfly Manual Bot (red error)
- Jogr Archive Bot (yellow warning)
- Sybil Archive Bot (green checkmark)

Each entry includes a timestamp indicating when the status was last updated. The main area shows a summary for "Integration 3" with the following counts:

- Errors: 0
- Warnings: 0
- Analysis: 0
- Passed Tests: 4

Below the summary, there is a message: "Added a new test case". A user profile for "kkeenan" is shown, along with the timestamp "Today at 2:26 PM" and "3 files modified". At the bottom, it indicates "2 Devices" with icons for an iPad and an iPod touch, both showing a green checkmark.

Document Revision History

This table describes the changes to *Xcode Continuous Integration Guide*.

Date	Notes
2013-10-22	Supplied instructions about adding user accounts to OS X Server.
2013-09-18	New document that provides guidelines and examples for using Xcode continuous integration features.



Apple Inc.
Copyright © 2013 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, AirPlay, Bonjour, Mac, OS X, Safari, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries.

Launchpad is a trademark of Apple Inc.

App Store is a service mark of Apple Inc.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.