
Swift Quick Start Guide

Source Control \$LastChangedRevision: 2810 \$

Abstract

The impatient may find the [Swift Really Quick Start Guide](#)¹ to be more convenient. This guide describes the steps needed to download, install, configure, and run the basic examples for Swift. If you are using a pre-installed version of Swift, you can skip directly to the [configuration section](#).

Table of Contents

1. Downloading a Swift Distribution	1
2. Installing a Swift Binary Package	1
3. Configuring Swift	2
4. Running Swift Examples	3

1. Downloading a Swift Distribution

There are three main ways of getting the Swift implementation: [stable releases](#), [nightly builds](#), and the [source code repository](#).

1.1. Stable Releases

Stable releases can be obtained from the Swift download page: [Swift Downloads Page](#)². Once you downloaded the package, please move to the [install section](#).

1.2. Nightly Builds

Swift builds and tests are being run every day. The [Swift downloads page](#)³ contains links to the latest build and test page. The nightly builds reflect a development version of the Swift code and should not be used in production mode. After downloading a nightly build package, please continue to the [install section](#).

1.3. Source Repository

Details about accessing the Swift source repository together with build instructions are available on the [Swift downloads page](#)⁴. Once built, the `dist/swift-svn` directory will contain a self-contained build which can be used in place or moved to a different location. You should then proceed to the [configuration section](#).

2. Installing a Swift Binary Package

¹ [reallyquickstartguide.php](#)

² <http://www.ci.uchicago.edu/swift/downloads/index.php#stable>

³ <http://www.ci.uchicago.edu/swift/downloads/index.php#nightly>

⁴ <http://www.ci.uchicago.edu/swift/downloads/index.php#nightly>

Simply unpack the downloaded package (`swift-<version>.tar.gz`) into a directory of your choice:

```
> tar -xzf swift-<version>.tar.gz
```

This will create a `swift-<version>` directory containing the build.

3. Configuring Swift

This section describes configuration steps that need to be taken in order to get Swift running. Since all command line tools provided with Swift can be found in the `bin/` directory of the Swift distribution, it may be a good idea to add this directory to your `PATH` environment variable:

```
> export PATH=/path/to/swift/bin:$PATH
```

3.1. Grid Security

For local execution of jobs, no grid security configuration is necessary.

However, when submitting jobs to a remote machine using Globus Toolkit services, Swift makes use of the [Grid Security Infrastructure \(GSI\)](#)⁵ for authentication and authorization. The requirements for this are detailed in the following sections. Note that GSI is not required to be configured for local execution (which will usually be the case when first starting with Swift).

3.1.1. User Certificate

GSI requires a certificate/private key pair for authentication to [Globus Toolkit](#)⁶ services. The certificate and private key should be placed into the `~/ .globus/usercert.pem` and `~/ .globus/userkey.pem` files, respectively.

3.1.2. Certificate Authorities Root Certificates

The Swift client libraries are generally required to authenticate the services to which they connect. This process requires the presence on the Swift submit site of the root certificates used to sign the host certificates of services used. These root certificates need to be installed in either (or both) the `~/ .globus/certificates` and `/etc/grid-security/certificates` directories. A package with the root certificates of the certificate authorities used in the [TeraGrid](#)⁷ can be found [heres](#).⁸

3.2. Swift Properties

A Swift properties file (named `swift.properties`) can be used to customize certain configuration aspects of Swift. A shared version of this file, `etc/swift.properties` in the installation directory can be used to provide installation-wide defaults. A per-user properties file, `~/ .swift/swift.properties` can be used for user specific settings. Swift first loads the shared configuration file and, if present, the user configuration file. Any properties not explicitly set in the user configuration file will be inherited from the shared configuration file. Properties are specified in the following format:

```
name=value
```

⁵ <http://www.globus.org/toolkit/docs/4.0/security/key-index.html>

⁶ <http://www.globus.org/toolkit>

⁷ <http://www.teragrid.org>

⁸ <http://security.teragrid.org/TG-CAs.html>

For details about the various properties Swift accepts, please take a look at the [Swift Properties Section](#)⁹ in the [Swift User Guide](#)¹⁰.

4. Running Swift Examples

The Swift examples can be found in the `examples` directory in the Swift distribution. The examples are written in the [SwiftScript language](#)¹¹, and have `.swift` as a file extension.

The Grid Security Infrastructure, which Swift uses, works with limited time certificates called proxies. These proxies can be generated from your user certificate and private key using one of **grid-proxy-init** or **cog-proxy-init** (the latter being a Java Swing interface to the former).

Execution of a Swift workflow is done using the **swift** command, which takes the Swift workflow file name as an argument:

```
> cd examples/swift
> swift first.swift
```

The [Swift Command Options Section](#)¹² in the [Swift User Guide](#)¹³ contains details about the various options of the **swift**.

⁹ <http://www.ci.uchicago.edu/swift/guides/userguide.php#properties>

¹⁰ <http://www.ci.uchicago.edu/swift/guides/userguide.php>

¹¹ <http://www.ci.uchicago.edu/swift/guides/userguide/language.php>

¹² <http://www.ci.uchicago.edu/swift/guides/userguide.php#swiftcommand>

¹³ <http://www.ci.uchicago.edu/swift/guides/userguide.php>