

## NAME

**notarytool** — Manage submissions to the Apple notary service.

## SYNOPSIS

```
notarytool submit file-path { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id | -p profile-name [--keychain keychain-path] } [--wait --no-s3-acceleration]  
  
notarytool info submission-id { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id | -p profile-name [--keychain keychain-path] }  
  
notarytool log submission-id { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id | -p profile-name [--keychain keychain-path] } [output-path]  
  
notarytool history { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id | -p profile-name [--keychain keychain-path] }  
  
notarytool wait submission-id { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id | -p profile-name [--keychain keychain-path] }  
  
notarytool store-credentials profile-name { -k key-path -d key-id -i issuer | --apple-id apple-id [--password app-specific-password] --team-id team-id } [--no-validate] [--keychain keychain-path | --sync]
```

## DESCRIPTION

Common subcommands include **submit**, **info**, **wait**, **history**, **log**, **store-credentials**, and **help**.

## BACKGROUND

Notarization gives users more confidence that the Developer ID-signed software you distribute has been checked by Apple for malicious components. Notarization is not App Review. The Apple notary service is an automated system that scans your software for malicious content, checks for code-signing issues, and returns the results to you quickly. If there are no issues, the notary service generates a ticket for you to staple (see stapler(1)) to your software; the notary service also publishes that ticket online where Gatekeeper can find it.

**notarytool** is a developer interface to this service. For example, **notarytool submit** [*options*] **--wait** *file-path* will verify *file-path* is one of the *Supported Upload File Formats*, initiate a connection with the Apple notary service, return the *Submission ID*, upload the file to the Apple notary service, wait for the submission to be processed by the Apple notary service, and exit when the processing is complete.

For more information on notarization, see the "Notarizing macOS Software Before Distribution" documentation at: <[https://developer.apple.com/documentation/xcode/notarizing\\_macos\\_software\\_before\\_distribution](https://developer.apple.com/documentation/xcode/notarizing_macos_software_before_distribution)>

## AUTHENTICATION OPTIONS

The following authentication options are available for all **notarytool** subcommands:

### App Store Connect API Keys

Developer ID team administrators can create App Store Connect API keys for the developers on their team by logging into <<https://appstoreconnect.apple.com/access/api>> and selecting the "Keys" tab. For security purposes, the private key can only be downloaded once.

**-k, --key** *key-path*

App Store Connect API key. File system path to the private key.

**-d, --key-id** *key-id*

App Store Connect API Key ID. For most teams this will be a 10 character alphanumeric string.

**-i, --issuer** *issuer*

App Store Connect API Issuer ID. The issuer ID is a UUID format string.

### App-specific Passwords

Create App-specific passwords by following the instructions on "Using app-specific passwords" at <https://support.apple.com/en-us/HT204397>. Any developer that has accepted the relevant agreements can use app-specific passwords with the Apple notary Service.

**--apple-id** *apple-id*

The Apple ID login username you use with Developer ID services.

[ **--password** *app-specific-password* ]

App-specific password for your Apple ID. You will be given a secure prompt on the command line if Apple ID and Team ID are provided but the **--password** option is not specified.

**--team-id** *wwdr\_team\_id*

The team identifier for the Developer Team to be used with this **notarytool** subcommand. Usually 10 alphanumeric characters. Your Apple ID may be a member of multiple teams, you can find Team IDs for teams you belong to by going to <https://developer.apple.com/account/#/membership>. You cannot get information on Submission IDs created by another *wwdr\_team\_id*.

The following options are available for all subcommands except **store-credentials**:

**-p, --keychain-profile** *profile-name*

Authenticate using credentials stored in the Keychain by **notarytool**. Use the profile name that you previously provided via the **store-credentials** command.

**--keychain** *keychain-path*

Pass the path to a keychain file to use for reading the keychain item specified by **-p** *profile-name*. If the specified keychain file is locked, you will be prompted to unlock it.

### SUPPORTED UPLOAD FILE FORMATS

**notarytool submit** works only with UDIF disk images, signed "flat" installer packages, and zip files. **notarytool** will do a shallow validation of the file before submission. Passing any other file format in *file-path* to **notarytool submit** will result in an error.

### SUBMISSION ID

**notarytool submit** returns a Submission ID as a UUID formatted string used to identify your submission. This Submission ID is necessary for the following subcommands: **info**, **wait**, and **log**. The Submission ID is also necessary when requesting support for most Apple notary service issues.

Submission IDs are unique to the development team that generated them. You can only retrieve information for submissions created by your team.

### COMMON OPTIONS

The following output control options are available for all **notarytool** subcommands:

**--version** Outputs the current version and build number of **notarytool**.

**-v, --verbose** Enable streaming of DEBUG level logs to stderr. This option can help the user decipher why a particular operation may have failed. Subsequent versions of **notarytool** may change the verbose logging. Do not write scripts assuming spe-

cific messages will continue to exist in the current form.

**--progress, --no-progress**

Display progress indicators. Only compatible with *normal* output format. **notarytool** defaults to **--progress**.

**-f, --output-format *format***

Desired output format. Note that *json* and *plist* are incompatible with **--progress** as all output will be suppressed until the subcommand has completed. Options are **normal**, **json**, or **plist**. **notarytool** defaults to *normal* output format.

## SUBCOMMANDS

**help** Display summary usage information. See **notarytool help *subcommand*** for more detailed help.

**store-credentials** *profile-name authentication\_options* [options]

Save authentication credentials for the Apple notary service to the default login keychain. If using **--key-path** to pass the file path of a private key, the contents of the private key are stored in the new keychain item and the private key file can be deleted.

*profile-name* is the name of the new keychain item to create. Passing in a previously saved profile name will cause the old keychain item to be overwritten.

Other options:

**--validate, --no-validate**

Verify that the authentication credentials are valid for use with the Apple notary service before saving them to the keychain and return an error otherwise. The default is **--validate**.

**--keychain *keychain-path***

Pass the path to a keychain file to use for writing the keychain item specified by *profile-name*. If the specified keychain file is locked, you will be prompted to unlock it. Cannot be specified if **--sync** is used.

**--sync** Make the new profile synchronize and accessible across devices with iCloud Keychain. Uses the "Local Items" keychain if iCloud Keychain is disabled. Cannot be specified if **--keychain** is used.

**submit** *file-path authentication\_options* [options]

Submit an archive to the Apple notary service. *file-path* must be a *Supported Upload File Format*.

Other options:

**--wait, --no-wait**

Rather than exiting after a successful upload, begin waiting for your submission to complete. The default is **--no-wait**. See the **wait** subcommand for more information.

**--timeout *duration***

If **--wait** is enabled, **notarytool** will exit after polling for the specified duration. Although **notarytool** exits, the submission will continue to be processed by the Apple Notary service. See the **wait** subcommand for more information.

**--s3-acceleration, --no-s3-acceleration**

Use S3 Transfer Acceleration for potentially faster uploads. The default is **--s3-acceleration**

**info** *submission-id authentication\_options* [options]

Get status information for a previous submission. *submission-id* is a Submission ID returned from a previous invocation of the **submit** subcommand.

**log** *submission-id authentication\_options* [*output-path options*]

Retrieve notarization log for a single completed submission as JSON. *submission-id* is a Submission ID returned from a previous invocation of the **submit** subcommand.

Use *output-path* to specify a path for the new notarization log file, otherwise the notarization log is printed to **stdout**.

**history** *authentication\_options* [options]

Get a list of previous submissions for your development team that were submitted via **notarytool**.

**wait** *submission-id authentication\_options* [options]

Wait for completion of a previous submission. *submission-id* is a Submission ID returned from a previous invocation of the **submit** subcommand.

Only return from **notarytool** once the Apple notary service has responded with a status of "Accepted", "Invalid", "Rejected", or if a fatal error has occurred during submission. This option replaces the need for polling from a script.

Other options:

[ **--timeout** *duration* ]

**notarytool** will exit after polling for *duration*. The Notary service will continue processing the submission even if the timeout is reached. *Duration* is an integer followed by an optional suffix: seconds 's' (default), minutes 'm', hours 'h'. For example, these values all set the timeout to an hour: *3600*, *3600s*, *60m*, *1h*.

## ALTOOL MIGRATION

In fall of 2023, **altool** will no longer be supported for notarization. Use **notarytool** instead. If you previously invoked **altool** like this:

```
altool --notarize-app -f path/to/app.pkg --primary-bundle-id com.example.myapp --apiKey 7UD13000
--issuerId 6bc36aee-c5c8-11ec-9d64-0242ac120001
```

Instead, use **notarytool** like this:

```
notarytool submit path/to/app.pkg --key path/to/AuthKey_7UD13000.p8 --key-id 7UD13000 --issuer
6bc36aee-c5c8-11ec-9d64-0242ac120001 --wait
```

## EXAMPLES

To store app-specific password credentials in the default keychain as a profile named MyGreatCompany-ASP:

```
notarytool      store-credentials      MyGreatCompany-ASP      --apple-id
mygreataccount@example.com --team-id K36BKF7T3D
```

To store App Store Connect credentials in the iCloud keychain as a profile named MyGreatCompany:

```
notarytool      store-credentials      MyGreatCompany      --key
~/private_keys/AuthKey_59GAB85EFG.p8      --key-id      59GAB85EFG
--issuer a04788a9-0819-478d-936f-6ff0fd860df5 --sync
```

To submit a disk image to the notarization service, using previously saved credentials while waiting for response:

```
notarytool submit MyDiskImage.dmg -p MyGreatCompany --wait
```

If the previous command returned a Submission ID of d0d37a38-dc80-4603-bca9-80705a49cbbd, download the log:

```
notarytool    log      d0d37a38-dc80-4603-bca9-80705a49cbbd    -p  
MyGreatCompany ~/Desktop/developer_log.json
```

## PERFORMANCE TUNING

**notarytool** has some options that are designed to allow a developer to tune some characteristics specifically for their network conditions.

**notarytool** submit . . [ **--s3-acceleration** | **--no-s3-acceleration** ]

If you experience performance or connectivity issues, please try disabling S3 Transfer Acceleration with **--no-s3-acceleration** when submitting files to the notary service.

## SEE ALSO

codesign(1), stapler(1), altool(1), spctl(8), syspolicyd(8)

## HISTORY

The **notarytool** command first appeared in Xcode 13