Using the Turnkey Commandline

Reference information for how to run Turnkey from RunUAT.bat using a commandline interface

Turnkey is an **Unreal AutomationTool (UAT)** script accessed from `RunUAT.bat`. While the tools to use Turnkey from inside Unreal Editor will be sufficient for most users, the commandline provides more detailed and advanced options for managing SDKs. This guide will show you how to access the Turnkey commandline, and provide a reference for its various options.

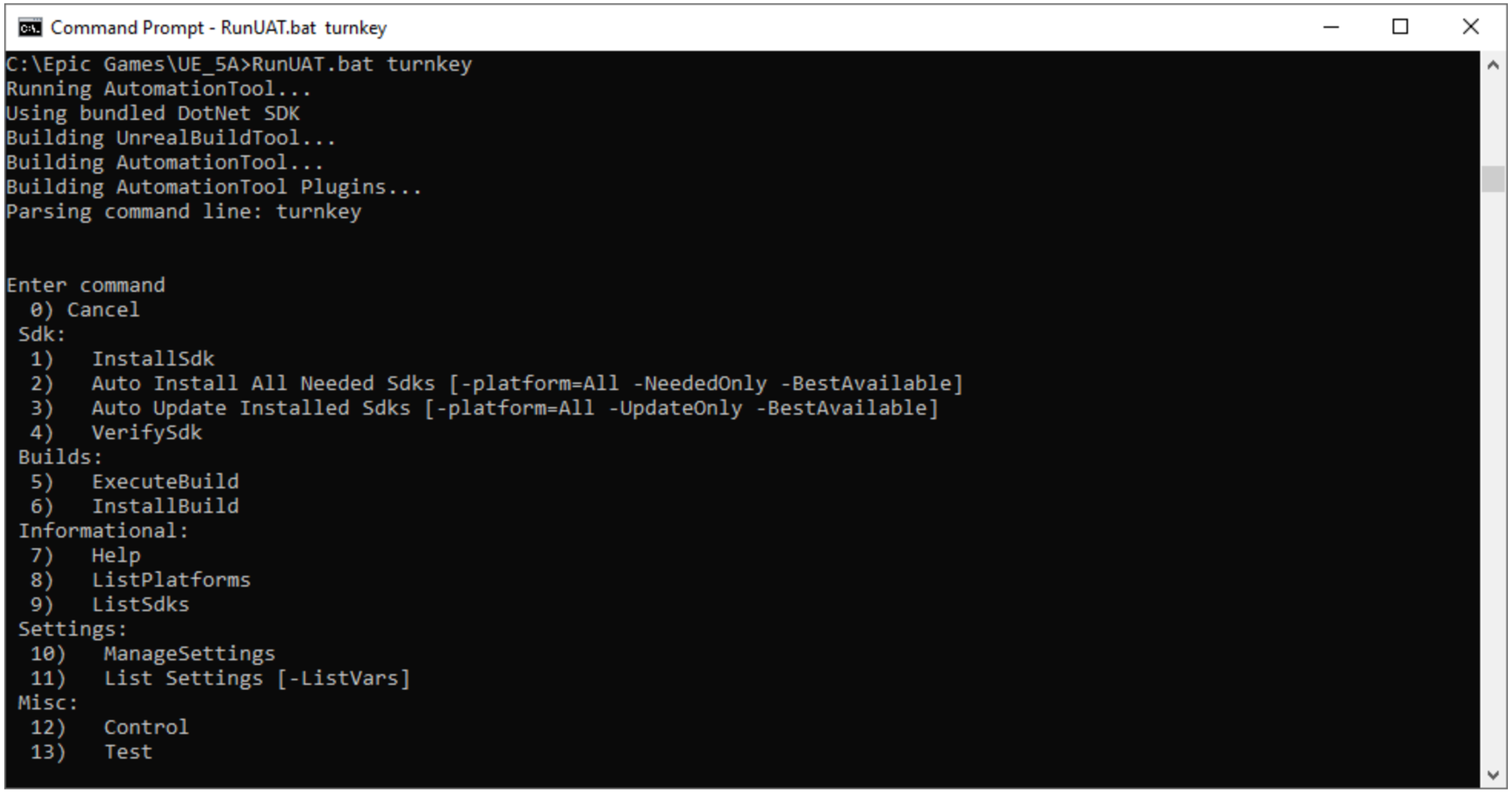
Accessing the Turnkey Commandline

To access Turnkey with a commandline, follow these steps:

1. Open your commandline of choice, such as the Windows command prompt.

2. Navigate to your Unreal Engine install directory.
3. Input `RunUAT.bat turnkey` to run Turnkey.

The commandline will spend a few moments building the AutomationTool, then start the Turnkey script and show a list of numbered commands.



From this menu, you can input the number corresponding to a command and press Enter to run it. Most of these commands will display a submenu with additional options specific to that command.

In all menus, entering **0** will cancel the current operation. If you choose to cancel in a submenu you will navigate back to the main Turnkey menu, while if you cancel in the main menu it will stop the script and exit. The sections below list the other available commands and their submenus.

Using Turnkey Commandline Arguments

Alternatively, when you run the `.bat` file, you can add a series of specifiers to skip these prompts and immediately run the command. Use the argument `-command=[command name]` to select a command, then provide additional specifiers to handle the other options.

For example, the following input would run the InstallSdk command, with the platform set to Android:

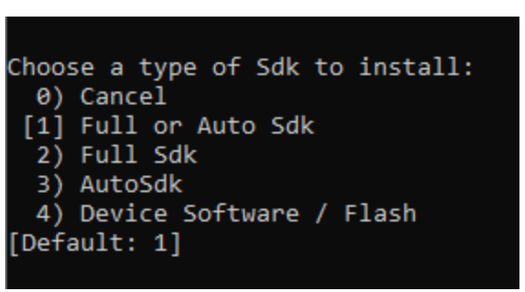
```
`RunUAT.bat turnkey -command=InstallSdk -platform=Android`

```

Copy full snippet

For information about the available specifiers for each command, refer to the sections below.

Installing SDKs

When you use the `InstallSdk` command, Turnkey will prompt you to select what SDK type you want to install.



The options are as follows:

1. `Full or Auto Sdk` will attempt to install either an AutoSDK or a Full SDK, and will select an AutoSDK if it is available.
2. `Full Sdk` will download a Full SDK available for your project, featuring the full array of components.
3. `AutoSdk` will attempt to install an AutoSDK, if it is available.
4. `Device Software / Flash` will download the most appropriate Flash SDK available for your project, featuring only components necessary for flashing dev kits and testing.

If Turnkey does not find your selected SDK type, it will abort the operation and throw an error.

After choosing your SDK type, Turnkey will also prompt you to select which platform's SDK you want to install.

Enter the number for the platform you want to set up, and Turnkey will initiate the download and installation process for that platform's SDKs. If no SDK is available the process will fail, provide an error message, and navigate back to the main menu.

Specifiers

The following specifiers are compatible when using `-command=InstallSdks` in the commandline.

Specifier	Description
`-Platform=`	Selects a platform. Use the name of the platform as seen in the prompt for platform selection. For example, `-Platform=Win64` would be valid, while `-Platform=Windows` would not. When you use this specifier, it will skip the platform selection submenu. `-Platform=All` will iterate through all available platforms.
`-NeededOnly`	Specifies that Turnkey should look for an AutoSDK as the SDK type.
`-BestAvailable`	Specifies that Turnkey should look for a Full SDK as the SDK type. When combined with -NeededOnly, it will look for either a Full SDK or an AutoSDK.
`-UpdateOnly`	Specifies that Turnkey should update an already installed SDK rather than perform the full installation.

The `Auto Install All Needed SDKs` command will run Turnkey with the specifiers `-command=InstallSdk -Platform=All -NeededOnly -BestAvailable`. This is equivalent to choosing a Full or Auto SDK and selecting **All of the Above** for your platform.

The `Auto Update Installed Sdks` command will run Turnkey with the specifiers `-command=InstallSdk -Platform=All -UpdateOnly -BestAvailable`.

Verifying SDKs

The VerifySdk command will prompt you to select the platform whose SDK you want to verify. Turnkey will then output information about the current SDK installation and check that it matches the parameters that Unreal Engine expects.

```
1
Installed Sdk validity:
Android: (Status=Valid, Installed=r21b, AutoSDK=, MinAllowed=r21a, MaxAllowed=r23a, Flags="InstalledSdk_ValidVersionExis
ts, Support_AutoSdk")
Scanning for envvar changes...
... done!
```

Specifiers

`-command=VerifySdk` is compatible with the `-Platform=` specifier.

Executing Builds

The `ExecuteBuild` command builds a project for a selected platform. Selecting this option will bring up a list of target platforms, followed by another prompt listing the available projects to build.

```
Choose a Normal project to execute, or s
n the commandline)
  0) Cancel
  1) [Show Templates Projects]
  2) [Show Sandbox Projects]
  3) [Show Misc Projects]
  4) [Show Samples Projects]
  5) [Show Collaboration Projects]
  6) [Show Personal Projects]
  7) [Show Recent Projects]
  8) QAGame
  9) [Browse...]
```

Projects are based on recognized `.uproject` names. For instance, `ShooterGame` is listed as a sample project. Once you select your platform and project, Turnkey will initiate the cooking and packaging process for your project.

Specifiers

`-command=ExecuteBuild` is compatible with the `-platform=` specifier. It can also use the `-project=` specifier to select a recognized project name and skip the prompt for selecting one. For example, the following input will attempt to build ShooterGame for the Win64 platform:

```
`RunUAT.bat Turnkey -command=ExecuteBuild -platform=Win64 -Project=Shootergame`
```

Copy full snippet

Installing Builds

> The functionality for installing builds with Turnkey is a work in progress in the Early Access build for UE5, and we are improving its reliability in the final release.

The `InstallBuild` command will bring up a list of previously created builds that are ready to install to devices, as well as a list of valid devices connected to your computer. Once you have selected both, Turnkey will install your build to the selected device.

Specifiers

`-command=InstallBuild` is compatible with the `-platform=` specifier. It can also use the `-device=` specifier. The format for devices is [Platform Type]@[Device Name], where the platform type is a platform recognized by Turnkey, and the device name is the ID of a device that is visible to your computer.

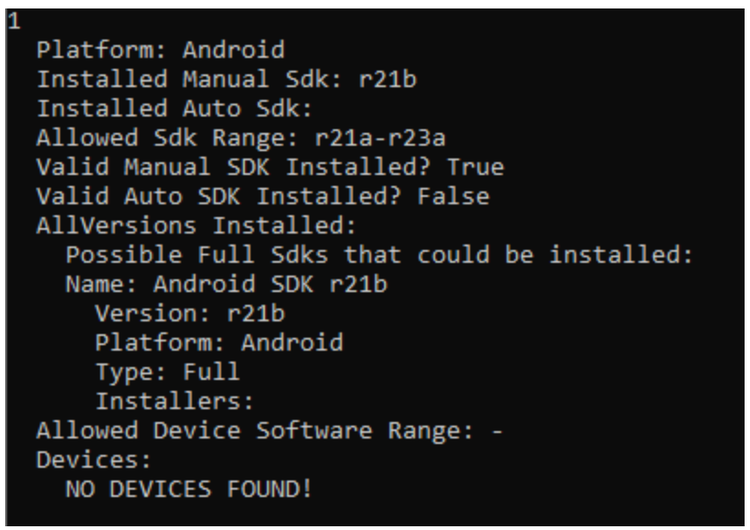
For example: -device=Android@ABCXYZ123. You can use ListPlatforms to see a list of devices and their IDs.

Help

The `Help` command opens a help menu that provides information about how to set up and use Turnkey. This includes information about how to format FileSource entries in `TurnkeyManifest.xml`, and the specific versioning formats for some platforms.

Listing Platform Information

The `ListPlatforms` command lists information about SDKs and devices set up for a selected platform. This includes information about SDK versions that are compatible with your current version of Unreal as well as visible devices on your network.



Specifiers

`-command=ListPlatforms` is compatible with the `-platform=` specifier.

List SDKs

The `ListSdks` command will output a list of the SDKs available in your FileSource repository. Turnkey will prompt you to select a platform you want to list SDKs for.

Specifiers

`-command=ListSdks` is compatible with the `-platform=` specifier.

Managing Settings

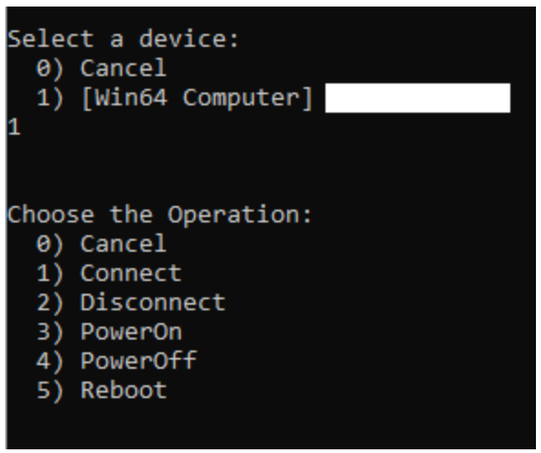
The `ManageSettings` command will display a series of variables you can configure. These are related to copy provider settings for your organization as well as credentials for specific platforms. These variables are normally located in multiple different files, such as `MobileProvision.ini` or your `TurnkeyStudioSettings.xml` file, but this command provides a centralized place to override them.

Specifiers

`-command=ManageSettings` is compatible with the `-ListOnly` specifier. This will output a list of all the variables you can configure, along with descriptions of what each of them does. The `List Settings` command in the Turnkey menu runs `-command=ManageSettings -ListOnly`.

Controlling Devices

The `Control` command brings up a menu that can control devices remotely. After you select a platform, it will display all of the devices matching that platform that are visible to your computer.



You can then turn the device on or off, reboot it, or connect/disconnect it. This functionality is the same as seen in the Device Manager inside the Unreal Editor.

Testing Turnkey

The `Test` command will run a diagnostic test to ensure that your environment is set up correctly. The test will attempt to connect to the copy provider of your choice and check for needed directories. If any part of the process fails, it will report errors.

```
Testing:

Enumerating googledrive:/SdkInstalls/*/DummyInstall.zip:
Connecting to GoogleDrive app 'Quickstart'. If this pauses here, check your web browser for an authentication tab. This
is required to be able to connect to Google Drive
Enumerating GoogleDrive spec: /SdkInstalls/*/DummyInstall.zip
Unable to find Drive named SdkInstalls
Enumerating googledrive:/SdkInstalls/*/*.xml:
Enumerating GoogleDrive spec: /SdkInstalls/*/*.xml
Unable to find Drive named SdkInstalls
Enumerating googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/*:
Enumerating GoogleDrive spec: '1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/*
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/Switch/
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/Apple/
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/Mac/
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/PS4/
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/DummyTurnkeyManifest.xml
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/DummyInstall.zip
  googledrive:'1O6V1JO-LNZYxcB6wIqCiHjQ44mz4tv7S'/Test2/
Enumerating file:D:\Engine\*\Build\*.xml:
Enumerating file:D:\Engine\*\*:
Enumerating perforce://UE4/Main/*/Build/*.xml:
Finding clientspec usable with //UE4/Main/*/Build/*.xml...
Unable to find a clientspec for the perforce operation //UE4/Main/*/Build/*.xml, looking for a depot client

Would you like to create one? [y/N]
[Default: N]
n
Skipping operation
Copying googledrive:/SdkInstalls/Installers/DummyInstall.zip:
```