

This topic provides step-by-step instructions for acquiring the LabKey Server source code, installing required components, and building LabKey Server from source. The instructions are written for a Windows machine; use in conjunction with the topic [Notes on Setting up OSX for LabKey Development](#), to set up development on a Mac or any OSX or Linux machine.

- [Checklist](#)
- [Obtain the LabKey Source Files](#)
- [Other Required Prerequisites](#)
 - [Java](#)
 - [Tomcat](#)
 - [Database](#)
- [Gradle Configuration](#)
- [Environment Variables and System PATH](#)
- [Install IntelliJ IDEA](#)
- [Build and Run LabKey](#)
- [Post-installation Steps](#)
- [Optional Modules on GitHub](#)
- [Install Optional Components](#)

Additional Topic:

- [Troubleshooting](#)

Checklist

A checklist, guiding you through the setup process, is available for download: [LabKey_Development_Server_Checklist.xlsx](#)

Obtain the LabKey Source Files

The LabKey source files are stored in two version control systems: (1) the build system and test sample data are stored in a Subversion (SVN) repository and (2) the core platform and all commonly distributed modules are stored in multiple GitHub repositories. To build LabKey Server, you need to checkout code from both version control systems.

Install TortoiseSVN

The following instructions apply to Windows machines. To install SVN on non-Windows machines see [Access the Source Code](#).

- [Download the latest version of TortoiseSVN](#).
- Install TortoiseSVN on your local computer.
- On the list of features to install, include the **command line client tools**.
- Add the TortoiseSVN/bin directory to your PATH (if it was not automatically added).

Checkout LabKey Source Files

- Create a new directory to hold the LabKey source files, for example, on Windows: C:\dev\labkey\trunk. This directory is referred to as <LABKEY_HOME> below.
- In Windows Explorer, right-click the new directory and select **SVN Checkout**.
- Enter the URL for the LabKey repository:

```
https://svn.mgt.labkey.host/stedi/trunk
```

- No username/password is required.
- Click **OK** to checkout the source files.

Install a Git Client

You can use GitHub Desktop or another Git client.

- Install GitHub Desktop:
 - GitHub for Windows <https://windows.github.com/>
 - GitHub for Mac <https://mac.github.com/>
- Install a command-line git option for your platform:
 - Git: <https://git-scm.com/downloads>

Clone Core Modules from GitHub

- Clone the following core repositories. Note that the first two go into <LABKEY_HOME>\server\modules, while the third goes into <LABKEY_HOME>\server.

```
<LABKEY_HOME>\server\modules> git clone  
https://github.com/LabKey/platform.git  
  
<LABKEY_HOME>\server\modules> git clone  
https://github.com/LabKey/commonAssays.git  
  
<LABKEY_HOME>\server> git clone  
https://github.com/LabKey/testAutomation.git
```

Install Required Prerequisites

Java

- [Download the most recent supported version of OpenJDK.](#)
- To install the JDK, unzip the archive to a chosen directory (for example, C:\labkey\apps\jdk-##).

To update the JDK in the future, you will:

- 1. repeat the steps above (installing the new version in a location parallel to the old)
- 2. update your system variable JAVA_HOME to point to the new location
- 3. update the IntelliJ > File > Project Structure > SDK > labkey home path to point to the new location. After changing the path, rename the new version "labkey".

Tomcat

- [Download the most recent supported release of Tomcat.](#) **Note:** Download a ZIP or TAR.GZ distribution, not the Windows Service Installer.
- To install Tomcat, unzip it to a chosen directory (for example, C:\labkey\apps\tomcat-#.###).

Install a Database

- Install one of the following database servers: PostgreSQL *or* Microsoft SQL Server. Find [supported versions in this topic](#).
- Take note of the username/password you select during installation.
- Platform-specific installation instructions:
 - **PostgreSQL on Windows:** Download and run the [Windows PostgreSQL one click installer](#).

- **PostgreSQL on Linux:** Install using your Linux machine's native tools or download a [PostgreSQL binary package](#). Follow the instructions in the downloaded package to install PostgreSQL.
- **MS SQL Server:** Follow the instructions in this topic: [Install Microsoft SQL Server](#). (But ignore the instructions around the config file labkey.xml, which do not apply to setting up a development server.)

Node.js and npm

The LabKey build depends on Node.js and the node package manager (npm). The build process automatically downloads and installs the versions of npm and node that it requires. You should not install npm and node yourself. For details on the Node.js dependency and build-prod script requirement, see [Node.js Build Dependency](#).

Gradle Configuration

Note that you do *not* need to install Gradle. LabKey uses the gradle wrapper to make updating version numbers easier. The gradle wrapper script (either gradlew or gradlew.bat) is included in the SVN sync and is already in the <LABKEY_HOME> directory.

Create a gradle properties file to contain information about global settings for your gradle build following these steps:

- Create a ".gradle" directory in your home directory.
 - On OSX and Linux, typically: /Users/<you>/.gradle.
 - On Windows, typically: C:\Users\<you>\.gradle). Note: the Windows file explorer may not allow you to create a folder beginning with a period. To solve this navigate to C:\Users\<you>\ in the command prompt and type mkdir .gradle.
- Create a gradle.properties file in the .gradle directory using the following process:
 - Copy the file <LABKEY_HOME>/gradle/global_gradle.properties_template to C:\Users\<you>\.gradle. Rename the file to create a file named gradle.properties.
 - In the gradle.properties file, substitute your <CATALINA_HOME> directory (the location of your Tomcat installation), including the specific version number, for the value after systemProp.tomcat.home. Use forward slashes, not backslashes, for the Tomcat path, even on Windows. For example:

```
systemProp.tomcat.home=C:/apache/tomcat-  
#.#.##
```

Environment Variables and System PATH

- **JAVA_HOME**

- Create or modify the system environment variable JAVA_HOME so it points to your JDK installation location (for example, C:\labkey\apps\jdk-##). Note: If you've already set the JAVA_HOME variable to point to your installation of the JRE, you should modify it to point to the JDK.

- **CATALINA_HOME**

- Create or modify the system environment variable CATALINA_HOME so that it points to your Tomcat installation (for example, C:\labkey\apps\tomcat-#.#.##).

- **PATH**

- Add the JDK to your system PATH. Using the JAVA_HOME variable will simplify upgrades later

```
%JAVA_HOME%\bin
```

- Add the following location to your system PATH. This directory won't exist yet, but add it to the path anyway.

```
<LABKEY_HOME>\build\deploy\bin
```

For example, C:\dev\labkey\trunk\build\deploy\bin.

OSX Example

On OSX, for example, you would place the environment variables in your .bash_profile:

```
export JAVA_HOME="/usr/libexec/java_home -v 1.11"  
export CATALINA_HOME=$HOME/apps/tomcat  
export LABKEY_HOME=$HOME/labkey/trunk  
export PATH=$LABKEY_HOME/build/deploy/bin:$PATH
```

GWT Browser Settings (*Optional*)

The default developer build is optimized for Chrome but the target browser can be controlled through either a command line parameter or by setting the 'gwtBrowser' property in the gradle.properties file. Available settings are: gwt-user-chrome (the default value), gwt-user-firefox, or gwt-user-ie.

Open the LabKey Project in IntelliJ

The LabKey development team develops LabKey using IntelliJ IDEA. You can use the license-free Community Edition if you are planning on modifying or extending the LabKey source code. Below we describe how to configure the IntelliJ development environment; we recommend employing the same general principles if you are using a different development environment. Some developers have experimented with Eclipse as the IDE and you can find some set up details on the [Developer Message Board](#).

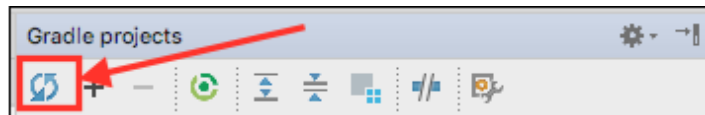
Install IntelliJ

- Download and install the latest version of [IntelliJ IDEA](#). Either the Community or Ultimate Editions will work.

Configure the LabKey Project in IntelliJ

- Create the workspace.xml file as follows:
 - Copy the file `<LABKEY_HOME>/.idea/workspace.template.xml`. Rename the copy to create a file named `<LABKEY_HOME>/.idea/workspace.xml`
 - This file configures the debug information for the LabKey project. To review the debug settings go to **Run > Edit Configurations** in IntelliJ.
- Open the LabKey project in IntelliJ:
 - Launch IntelliJ.
 - If your IntelliJ install is brand new, you will see the "Welcome to IntelliJ" pop up screen. Click **Open**.
 - If you have previously installed IntelliJ, select **File > Open**.
 - Select the LabKey IntelliJ project directory:
`<LABKEY_HOME>`
 - If asked about an "Unlinked Gradle project", DO NOT "Import Gradle project" in the default way from IntelliJ. See the troubleshooting section [Starting Over with Gradle + IntelliJ](#) for more information.
- Set **CATALINA_HOME**
 - Select **File > Settings > Appearance & Behavior > Path Variables**. (On a Mac, the menu path is **IntelliJ IDEA > Preferences > Appearance & Behavior > Path Variables**).
 - Click the plus icon in the upper right. Set the CATALINA_HOME path variable to `<CATALINA_HOME>`, the root directory of your Tomcat installation, for example, `C:\labkey\apps\apache\apache-tomcat-#.##.##`.

- Click **OK** to close the Settings window.
- Configure the Target JDK
 - In IntelliJ, select **File > Project Structure**.
 - Under **Project Settings**, click **Project**.
 - Under **Project SDK** click **New** and then click **JDK**.
 - Browse to the path of your JDK, for example, (C:\java\jdk-##), and click **OK**.
 - Click **Edit**. Change the name of the JDK to "labkey".
 - Click **OK** to close the Project Structure window.
- Open the Gradle tool window at **View > Tool Windows > Gradle**.
 - Click the **refresh** icon. This will take as much as 5-20 minutes. You should start seeing messages about its progress. If not, something is probably hung up. Wait for this sync to complete before progressing with further IntelliJ configuration steps.



- Edit configuration options as follows:
 - Select **Run > Edit Configurations**. (If the menu is greyed-out, wait until IntelliJ finishes indexing the project files.)
 - Open the **Application** node in the left panel and select **LabKey Dev**.
 - VM options: Confirm that the path separators are appropriate for your operating system. On Windows, ensure that the paths to the jar files are separated by semicolons. For example: ".\bin\bootstrap.jar;.\bin\tomcat-juli.jar;C:/Program Files (x86)/JetBrains/IntelliJ IDEA 2016.3.3/lib/idea_rt.jar". For Macs, the paths should be separated by a colon.
 - Confirm that **Working Directory** points to your current Tomcat installation (i.e. to CATALINA_HOME).
 - Confirm that the dropdown labeled **Use classpath of module** is set to **api_main** or **org.labkey-api_main** (whichever is available).
 - Click **OK**, to close the Run/Debug Configurations window.
- Be sure that IntelliJ has enough heap memory. The default max is OK if you're just dealing with the core modules, but you will **likely need to raise the limit** if you're adding in customModules, optionalModules, etc. 3GB seems sufficient.

Build and Run LabKey

Configure the Appropriate .properties File

The LabKey source includes two configuration files, one for use with PostgreSQL (*pg.properties*) and one for use with Microsoft SQL Server (*mssql.properties*), each specifying JDBC settings, including URL, port, username, password, etc.

- If using PostgreSQL, open the file
 <LABKEY_HOME>/server/configs/pg.properties
- If using MS SQL Server, open the file
 <LABKEY_HOME>/server/configs/mssql.properties
- Edit the appropriate file, adding your values for the **jdbcUser** and **jdbcPassword**. (This password is the one you specified when installing PostgreSQL or MS SQL Server. If your password contains an ampersand or other special XML characters, you will need to escape it in the .properties file, as the value will be substituted into an XML template without encoding. For example, if your JDBC password is "this&that", then use the escaped version "this&that".)

Run pickPg or pickMSSQL

- In a command window, go to the directory <LABKEY_HOME>
- Run "gradlew pickPg" or "gradlew pickMSSQL" to configure labkey.xml with the corresponding database settings.
- You may need to manually create the directory
 <CATALINA_HOME>/conf/Catalina/localhost.

When you build LabKey, the values that you've specified in the *mssql.properties* or *pg.properties* file are copied into the LabKey configuration file, labkey.xml, overwriting previous values. This file is then copied into
 <CATALINA_HOME>/conf/Catalina/localhost.

Build LabKey

To learn more about the build process, the various build targets available, and how the source code is transformed into deployed modules, see [Build LabKey from Source](#).

- On the command line, go to the <LABKEY_HOME> directory, and invoke the gradle build target:

```
gradlew deployApp
```

To control which modules are included in the build, see [Customize the Build](#).

Run LabKey Server

To run and debug LabKey:

- Select **Run > Debug 'LabKey Dev'** in IntelliJ.
- If Tomcat starts up successfully, navigate your browser to <http://localhost:8080/labkey> to begin debugging (assuming that your local installation of Tomcat is configured to use the Tomcat default port 8080).

While you are debugging, you can usually make changes, rebuild, and redeploy LabKey to the server without stopping and restarting Tomcat. Occasionally you may encounter errors that do require stopping and restarting Tomcat.

Post-installation Steps

Install R

- [Install and configure the R programming language](#)

Run the Basic Test Suite

- Run the command within your <LABKEY-HOME> directory, to initiate automated tests of LabKey's basic functionality:

```
gradlew :server:testAutomation:uiTest -Psuite=DRT
```

Note that 'R' [must first be configured](#) for these tests to run. Other automated tests are available. For details, see [Run Automated Tests](#).

Optional Modules on GitHub

Many optional modules are available from the [LabKey repository on GitHub](#). To include these modules in your build, install a Git client and clone individual modules into the LabKey Server source.

Clone Modules from LabKey's GitHub Repository

- To add a GitHub module to your build, clone the desired module into trunk/labkey/server/optionalModules. For example, to add the 'workflow' module:

```
C:\svn\trunk\server\optionalModules>git clone  
https://github.com/LabKey/workflow.git
```

Note that you can get the URL by going to the module page on GitHub (for example, <https://github.com/LabKey/workflow>), clicking **Clone or Download**, and copying the displayed URL.

Manage GitHub Modules via IntelliJ

Once you have cloned a GitHub module, you can have IntelliJ handle any updates:

To add the GitHub-based module to IntelliJ (and have IntelliJ generate an .iml file for the module):

- Edit your settings.gradle file to [include the new module](#)
- In IntelliJ, open the Gradle tool window at **View > Tool Windows > Gradle**.
- Refresh the Gradle window by clicking the **arrow circle** in the upper left of the Gradle window

To update the GitHub-based module using IntelliJ:

- To have IntelliJ handle source updates from GitHub, go to **File > Settings** (or **IntelliJ > Preferences**).
- Select **Version Control**.
- In the **Directory** panel, scroll down to the **Unregistered roots** section, select the module, and click the **Plus** icon in the lower left.
- In the **Directory** panel, select the target module and set its VCS source as **Git**, if necessary.
- Note that IntelliJ sometimes thinks that subdirectories of the module, like module test suites, have their sources in SVN instead of Git. You can safely delete these SVN sources using the **Directory** panel.
- To sync to a particular GitHub branch: in IntelliJ, go to **VCS > Git > Branches**. A popup menu will appear listing the available Git modules. Use the popup menu to select the branch to sync to.

If you have added a new module to your enlistment, be sure to [customize the build](#) to include it in your Gradle project and then refresh the Gradle window to incorporate it into IntelliJ, as described above.

Install Optional Components

Mass Spec and Proteomics Tools

LabKey Server's mass spectrometry and proteomics binaries are provided as a separate (and optional) enlistment. To add these binaries, follow the instructions in the topic: [Add the Proteomics Binaries](#)

Related Topics

- [Tutorial: Hello World Module](#)
- [Access the Source Code](#)

- [Build LabKey from Source](#)
- [Customize the Build](#)
- [SVN and Git Ignore Configurations](#)
- [Build Offline](#)
- [Gradle Cleaning](#)
- [Gradle Properties](#)
- [Gradle: How to Add Modules](#)
- [Gradle: Declare Dependencies](#)
- [Gradle Tips and Tricks](#)
- [Run Selenium Tests](#)
- [Create Production Builds](#)
- [Gradle Build Overview](#)
- [Node.js Build Dependency](#)
- [Machine Security](#)
- [Notes on Setting up OSX for LabKey Development](#)
- [Troubleshoot Development Machines](#)

☆ **Premium Resource Available**

Subscribers to premium editions of LabKey Server can learn more about using IntelliJ in this topic

- **Premium Resource: IntelliJ Reference**

[Learn more about premium editions](#) 