

Ignition! v7.7 Upgrade Guide



Thinking of upgrading your existing Ignition system to version 7.7? Can't wait to get your hands on all those new features? Great! But first, read this short upgrade guide to be prepared.

1. Make a Backup First:

Before any major upgrade it is important to make a backup of your existing system, and write down what version you were using. That way, if you encounter any unexpected turbulence, you can revert back quickly.

2. Report Issues:

If you find anything out of place, let us know! Visit the Ignition 7.7 forum to report any anomalies you find.

3. Call your Sales Rep:

Ignition 7.7 is a major upgrade to the Ignition platform and all of its modules. When you upgrade, all of your modules will revert back to trial mode until you update your license.

- a. If you have Ignition Total Care Upgrade Protection, you don't need to pay anything additional to upgrade your license key, but you will need to update the license that is installed on your server(s). If you have internet access on your gateway, this is as easy as clicking the "Update License" link on the license page of the gateway's configuration web interface. If you don't have internet access, you'll need to do an offline activation.
- b. If you don't have Total Care Upgrade Protection, you'll need to contact your sales rep and arrange to purchase the upgrade.

4. Upgrade Java:

Ignition 7.7 requires Java SE 8 for the Gateway. Download Java 8 here:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Clients and Designers still support Java 6, 7, and 8.

5. Custom Tomcat settings:

If you added items like MIME types to `tomcat/conf/web.xml` folder by hand, you'll need to apply those changes to `webserver/webdefault.xml` after upgrading.

6. Scripting Changes:

Ignition 7.7 has a number of new scripting features. If you're a heavy script user, you should be aware of what these changes mean.

a. Going forward, "app.*" is being replaced by "project.*".

However, if you upgrade an existing installation with `app.*` scripts, they will remain indefinitely. Whether you move them over to the `project.*` space is up to you.

b. New scripting locations use new scoping rules.

Ignition's scripting semantics have changed somewhat for new scripts. We now correctly use python's module scoping rules. This has two major effects.

- i. Variables are now directly accessible from within functions. For example:

Old Scoping

```
x = 5
def myFunction(x = x):
    print x
```

New Scoping

```
x = 5
def myFunction():
    print x
```

- ii. The `global` keyword now works differently. Instead of making the variable truly "global", it declares it global within the scope of the current module. See the Python Programming FAQ for more details. In order to make new scripts compatible with older scripts that used global variables, there is a new function: `system.util.getGlobals()` which provides access to the old global space.