

Migration Guidance

The Apache NiFi community recognizes how important it is to provide reliable releases on a number of levels. But one of the most important aspects is that we consider the importance of changes which create new behavior, change existing behavior and so on. We're committed to being a responsible community whereby we can continue to evolve the capabilities and features of NiFi and users can have a well understood and reliable upgrade path. We're committed to ensuring backward compatibility issues are made quite rare or that the impact is clearly understood, minimized, and communicated. You can read more about [our approach to version management](#). If you find that we've violated this commitment in any way please send us an email at dev@nifi.apache.org and we'll work to resolve it for you and other users.

To summarize:

- When moving between patch (also known as incremental) version changes such as 0.1.0 to 0.1.1 users should be safe to assume a clean upgrade can occur with no risk of behavior changes other than bug fixes and no compatibility issues.
- When moving between minor changes such as 0.1.0 to 0.2.0 users can expect new behaviors and bug fixes but backward compatibility should be protected.
- When moving between major changes such as 0.x.y to 1.0.0 there may be backward compatibility impacting changes largely focused on removal of deprecated items.

The following guidance is specific to the indicated version changes. It will contain specific items that users should be aware of when moving between versions:

- Migration to 0.5.0
 - The Proxy permissions were made more granular and now support specific permissions for Read, Write, and Delete. Any existing users that were granted the Proxy permission will end up with only Write permissions and will need to be granted Read and Delete. This primarily impacts the users that represent NiFi instances which will need Read and Write permissions to Proxy, in order to perform operations on behalf of the end users in NiFi.
- Migration to 0.4.0
 - A new provider must be defined in providers.xml for extension bundles. Compare the providers.xml that comes with the release to your existing providers.xml and update accordingly.
 - A new optional config file was added called registry-aliases.xml, add this to your conf directory if desired and see nifi-registry.properties for how to reference it.
 - New optional properties for transforming identities to lower/upper case were added to nifi-registry.properties, compare your existing file to the one from the release and update accordingly.
- Migrating from 0.2.0 to 0.3.0
 - The 0.3.0 release introduces a new, optional RangerAuthorizer provider as a NiFi Registry extension. If you wish to migrate from an existing ManagedAuthorizer backed by a FileAccessPolicyProvider, you will have to recreate all access policy rules in the externally configured Apache Ranger instance. Please see the [nifi-ranger extension README file](#) for full documentation on how to install and configure the extension.
- Migrating from 0.1.0 to 0.2.0
 - The 0.2.0 release introduces new database configuration in *nifi-registry.properties* which provides more flexible options. When upgrading from 0.1.0, ensure that *nifi.registry.db.directory* points to the location of the existing H2 database to ensure it is migrated to the new database. Please consult the [Database Properties](#) section of the Admin Guide for further details of these properties.
- Migrating metadata database for 0.2.0 (H2 to PostgreSQL)
 - By default, 0.2.0 uses H2 for storing metadata. If this default is used, and then a migration to PostgreSQL is desired, refer to the below steps:

1. Create a database in PostgreSQL (For example, nifiregistry)

```
postgres=# CREATE DATABASE nifiregistry;
```

2. Update *nifi-registry.properties*. Modify PostgreSQL configuration values as per the environment. The *nifi.registry.db.directory* points to the location of the existing H2 database.

```
nifi.registry.db.directory=./database
nifi.registry.db.url.append=;LOCK_TIMEOUT=25000;WRITE_DELAY=0;AUTO_SERVER=FALSE

# PostgreSQL configuration
nifi.registry.db.url=jdbc:postgresql://localhost/nifiregistry
nifi.registry.db.driver.class=org.postgresql.Driver
nifi.registry.db.driver.directory=/Library/PostgreSQL/pgJDBC
nifi.registry.db.username=postgres
nifi.registry.db.password=postgres
nifi.registry.db.maxConnections=5
nifi.registry.db.sql.debug=false
```

3. Update the H2 database file name to be `nifi-registry.mv.db`

a) Navigate to location of existing H2 database.

b) Rename file.

```
$ mv nifi-registry-primary.mv.db nifi-registry.mv.db
```

4. Restart the Registry.