XNAT Installation Guide

XNAT 1.7 is different from previous versions of XNAT:

- **xnat_builder** is gone
- There's no need to run setup or update scripts
- It can be deployed directly into Tomcat with only minimal changes to the Tomcat configuration

Don't be alarmed by the lack of many of the steps previously required to get XNAT going! You're doing the right thing!

Quick Installation

This quick installation guide does not describe how to configure a front-end proxy such as nginx or Apache HTTPD for your server. Production configurations almost always use a front-end proxy for handling the standard HTTP/HTTPS ports 80 and 443, SSL certificate management, and other functions. If you haven't configured a front-end proxy and are using the standard Tomcat configuration, you will need to add the Tomcat listener port to your server URL. For example, the address would be http://server:8080 rather than http://server.

If you're familiar with installing applications in Tomcat and configuring services in your server environment, this quick installation includes just the basic steps required to get XNAT up and running. More details can be found in the Comprehensive Installation section below.

1. Install all prerequisites: Java JDK 1.7 or 1.8, Apache Tomcat 7, and PostgreSQL 9.1 to 9.6 (9.6 strongly recommended)
2. Create XNAT's data folders: **archive**, **build**, **cache**, **ftp**, and **prearchive**. These folders must be owned by the user that owns Tomcat. The standard XNAT practice is to put these folders in one location, usually named **/data/xnat** (or, if you have a custom Tomcat user, that user's name, e.g. **/data/foo**), so you end up with **/data/xnat/archive**, **/data/xnat/build**, and so on.
3. Create a PostgreSQL user and database for your XNAT installation:

   Creating XNAT PostgreSQL user and database

   ```
   $ sudo su - postgres -c "createuser -D xnat"
   $ sudo su - postgres -c "createdb -O xnat xnat"
   $ sudo su - postgres -c "psql -c \"ALTER USER xnat WITH PASSWORD 'xnat'\""
   ```

   There are different ways to create users and databases, as well as manage how users access the databases. For some configurations, you may not need to set a password for the user at all. And as long as you are accessing the database from XNAT on the same server, you may not need to change any PostgreSQL configuration options at all. See Configuring PostgreSQL for XNAT for more information.
4. Create the XNAT home folder. This is where the XNAT service stores and accesses configuration files, logs, plugins, and temporary files. That means this folder must be writable by the Tomcat user. If you create a custom user, you can just use the home folder for this user as your XNAT home.
5. Configure the property **xnat.home** in your Tomcat configuration file as part of the **JAVA_OPTS** environment variable to indicate the location of the XNAT home folder you created in the previous step. The location of the Tomcat configuration file depends on the platform:
   - On Debian/Ubuntu it is **/etc/default/tomcat7**
   - On Fedora/Red Hat, including CentOS 6, it is **/etc/tomcat7/tomcat7.conf**
   - On CentOS 7.3 (and perhaps earlier CentOS 7 versions), it is **/etc/tomcat/tomcat.conf**

   Running XNAT on Windows is no longer supported and recent versions of XNAT have not been tested on Windows. We recommend either using a different operating system or running XNAT in a Vagrant virtual machine.
6. You need to append the value for **xnat.home** to either the CATALINA_OPTS or JAVA_OPTS environment variable value:

   Setting xnat.home

   ```
   CATALINA_OPTS="${CATALINA_OPTS} -Dxnat.home=/data/xnat/home"
   ```

   See the section below on Tomcat for more information about configuring these variables.
7. Create four folders under the XNAT home folder: **config**, **logs**, **plugins**, and **work**. As with the XNAT home folder, these must be writable by the Tomcat user.
8. In your newly created **config** folder, create the file **xnat-conf.properties**. You can use the default version of this file as a starting point. Set the database and other properties to the appropriate values.
9. Download the latest release of the XNAT war and pipeline installer to your server:

Creating XNAT PostgreSQL user and database

```
$ curl -L -o xnat-web-1.7.4.1.war https://api.bitbucket.org/2.0/repositories/xnatdev/xnat-web/downloads/xnat-web-1.7.4.1.war
$ curl -L -o xnat-pipeline-1.7.4.zip https://api.github.com/repos/NrgXnat/xnat-pipeline-engine/zipball/1.7.4
```

10. Copy the war file into the Tomcat 7 **webapps** folder. The location of this folder depends on where Tomcat is installed:
    - **/var/lib/tomcat7/webapps** on Debian/Ubuntu-based systems
    - **/usr/share/tomcat7/webapps** on Fedora/Red Hat-based systems

    There is no need to restart the Tomcat service: when a war file is placed in the **webapps** folder, Tomcat automatically deploys the application to the context ***filename*** (if a file of the same name already exists, Tomcat will redeploy the application). That is, if you copy the war file in as **xnat-web-1.7.4.1.war**, your newly deployed XNAT application will be available at the URL http://server/xnat-web-1.7.4.1. If you'd like XNAT to be the root application, you should rename it to **ROOT.war**:

    Copying the XNAT war as the root application

    ```
    $ cp xnat-web-1.7.4.1.war /var/lib/tomcat7/webapps/ROOT.war
    ```

    You can monitor the start-up progress of your Tomcat server through the Catalina start-up log. On most Linux platforms, you can use **tail** for this purpose:

    Monitoring Tomcat Start

    ```
    $ tail -f /var/log/tomcat7/catalina.out
    ```

    You may need to use **sudo** if your user account doesn't have access to the folder or log file.
11. Install the XNAT 1.7 pipeline engine.
12. Enter the address of your XNAT server in a web browser, e.g. http://oldschool.edu. Wait for the login page to appear.
13. Log into XNAT with the username **admin** and password **admin**. You should be directed to the XNAT site setup page.
14. Enter all the required data in the site setup form and click the **Save** button at the bottom of the page.

If everything went well, you should now be logged into your new XNAT instance!

Comprehensive Installation

Prerequisites

To install XNAT 1.7, you'll need the following services installed and configured:

- Java JDK
- Apache Tomcat
- PostgreSQL

The particular versions required and any custom configuration is detailed in the following sections.

Java JDK

XNAT 1.7 was developed and tested extensively with Java 7. Recent releases of XNAT have also been tested extensively with Java 8, and this is the version that we currently recommend.

> ⚠ Java 9 and later versions of Java have not been well tested with XNAT, so are not currently supported.

You can download the Java JDK for Windows, OS X, and most Linux platforms from the Oracle Java SE Development Kit 8 Downloads page. You can install OpenJDK on most Linux platforms using the appropriate package manager application.

Apache Tomcat 7

XNAT 1.7 was developed and tested exclusively on Tomcat 7. We have had some success running under Tomcat 8, but there are known issues with some combinations of different versions of the JDK and Tomcat 8 that we have not been able to fully pin down. Running under Tomcat 8 is possible but is at your own risk until we can fully test and quantify the issues on that platform.

XNAT 1.7 is *not compatible* with Tomcat 6 and will not function at all in that environment! Tomcat 7 introduced support for the Java Servlet 3.0 specification. Many of the new features and improvements in XNAT 1.7 are based on Servlet 3.0 changes.

There are a few configuration changes to prepare Tomcat for XNAT 1.7. Only one of these is required:

- You *must* define the system variable **xnat.home** for Tomcat. The easiest way to do this is to modify the primary Tomcat configuration file:
    - For Debian and Ubuntu systems, this is located in **/etc/default/tomcat7**.
    - For Fedora, Red Hat, CentOS, and similar systems, this is located in **/etc/tomcat7/tomcat7.conf** or **/usr/share/tomcat7/conf/tomcat7.conf** (these are the same file: **/usr/share/tomcat7/conf** is a symlink to **/etc/tomcat7**).
    - OS X doesn't have the same standard method of configuring properties for Tomcat as a service. Generally, you can create a file in the **bin** folder for your Tomcat installation named **setenv.sh**. You should also make the file executable with the command **`chmod +x setenv.sh`**. You can modify this file in the same way as the configuration files for the Linux platforms.
- You may want to change the Tomcat service user and group. These can be configured in the primary Tomcat configuration file as well. If you do change the user, you *must* change the ownership or permissions on all of the folders within the Tomcat installation. Failure to do this will result in permissions issues where the Tomcat service can't access its own configurations or application folders or even be able to write log files describing what went wrong. The general practice with the XNAT development and operations teams is to change the Tomcat user and group to something relevant to the project and application installation, e.g. our development servers are usually set up with the user set to **xnat** or **xnatdev**. See below for tips on changing the Tomcat user and group.
- Whether you change the user or group for Tomcat or not, all XNAT data folders must be owned by the Tomcat user or at least assigned to the same group as the Tomcat user with full privileges for that user or group. Again, with XNAT development, we have a folder structure **/data/xnat** that is owned by the user **xnat** and the group **xnat**. Everything under there is then owned by the same user so that Tomcat will be able to create and write to any folder under that structure.
- If you will be debugging code in your XNAT service, you can add the appropriate debug flags to the Java configuration along with the **xnat.home** setting.
- You can use the Tomcat manager application to manage your XNAT application within the Tomcat container. This is a relatively limited manage function, restricted to stopping, starting, undeploying (i.e. uninstalling), and deploying (i.e. installing) applications in the Tomcat container. However, this can be very useful if you're frequently installing or updating plugins in your XNAT application. In this case, you need to make sure the install and configure the Tomcat manager application. Note the need for separate roles and users for the **manager-gui**, **manager-script**, and **manager-jmx** roles. If you install the Tomcat manager application, these users *must* be configured appropriately or you won't be able to properly access the application to manage your container instance.

Configuring XNAT Home

Configuring the **xnat.home** folder in your Tomcat configuration should look something like this:

```
CATALINA_OPTS="${CATALINA_OPTS} -Dxnat.home=/data/xnat/home"
```

If you want to add the debugger flags, it would look like this:

```
CATALINA_OPTS="${CATALINA_OPTS} -Dxnat.home=/data/xnat/home -agentlib:jdwp=transport=dt_socket,server=y,
suspend=n,address=8000"
```

Changing the Tomcat User and Group

All of the Tomcat folders needed to owned (or at least fully accessible, including write access in most cases) by the user indicated by the **TOMCAT_USER** property. In most default installations, this user is named the same as the service (**tomcat7**, **tomcat**, etc.) and all of the folders are owned appropriately. If you change the **TOMCAT_USER** setting, you need to change the owner accordingly (the same thing goes for changing the group). This is complicated by the fact that the packaged installations of Tomcat on Debian- and Fedora-based systems actually install parts of the Tomcat application in different folders, then “integrate” these parts through symlinks (located in **/var/lib/tomcat7** on Debian-based systems and **/usr/share/tomcat7** on Fedora-based systems). Changing ownership just through the integrated installation folder doesn’t usually suffice to change permissions on folders that are symlinked. This can lead to confusing messages about not being able to create folders, write to files, etc.

On Debian- and Fedora-based Linux systems, you can change ownership for all of the folders quickly with these two commands:

```
# chown -RH --dereference xnat.xnat /var/lib/tomcat7
# chown --no-dereference xnat.xnat /var/lib/tomcat7/*
```

These commands are slightly different for CentOS 7:

```
# chown -RL --dereference xnat.xnat /usr/share/tomcat7
# chown --no-dereference xnat.xnat /usr/share/tomcat7/*
```

To verify the results of these operations, you can explicitly display the folders (this is for Ubuntu, references to **/var/lib/tomcat7** may need to be changed on Fedora-based systems):

```
# ls -ld /var/lib/tomcat7 /var/lib/tomcat7/* /etc/tomcat7 /var/log/tomcat7 /var/cache/tomcat7
drwxr-xr-x 4 xnat xnat 4096 Dec 6 22:41 /etc/tomcat7
drwxr-x--- 3 xnat xnat 4096 Dec 6 22:42 /var/cache/tomcat7
drwxr-xr-x 6 xnat xnat 4096 May 28 2016 /var/lib/tomcat7
drwxr-xr-x 3 xnat xnat 4096 May 28 2016 /var/lib/tomcat7/common
lrwxrwxrwx 1 xnat xnat 12 Jun 19 2015 /var/lib/tomcat7/conf -> /etc/tomcat7
lrwxrwxrwx 1 xnat xnat 17 Jun 19 2015 /var/lib/tomcat7/logs -> ../../log/tomcat7
drwxr-xr-x 3 xnat xnat 4096 May 28 2016 /var/lib/tomcat7/server
drwxr-xr-x 3 xnat xnat 4096 May 28 2016 /var/lib/tomcat7/shared
drwxrwxr-x 3 xnat xnat 4096 Dec 6 22:42 /var/lib/tomcat7/webapps
lrwxrwxrwx 1 xnat xnat 19 Jun 19 2015 /var/lib/tomcat7/work -> ../../cache/tomcat7
drwxr-x--- 2 xnat xnat 4096 Dec 8 20:14 /var/log/tomcat7
```

There are many other options you can add here, including settings for memory allocation, garbage collection, performance profiling, etc., but understanding and tuning these appropriately can be very detailed. These are covered in the Advanced Topics section of this administrator's guide.

PostgreSQL

XNAT 1.7 can run under PostgreSQL on versions as early as 9.1. However, much new development on XNAT, including plugins that are compatible with XNAT 1.7 will require PostgreSQL 9.4 or later. With that in mind, we strongly recommend installing XNAT 1.7 on 9.4 or later from the start.

> The PostgreSQL wiki has detailed documentation on installing the server on many different platforms .

Once you've installed PostgreSQL, you should create a user and database exclusively for XNAT. The XNAT standard practice is to name the database and user the same as the Tomcat user. There is no real link formed by these names, but it can reduce confusion. Also, the PostgreSQL client command-line utilities assume they should use the current user's name for both the database and user names when invoked. Thus, if you're operating as the user **xnat** , typing **psql** (without specifying a user with the **-U** parameter or a database by adding it to the command line) automatically tries to connect you to the database **xnat** as the database user **xnat**.

There are a number of different ways XNAT can connect to the database. How you configure this depends on your requirements for authentication and security, distribution of services (e.g. using another server to host your database), and so on. See Configuring PostgreSQL for XNAT for more information on setting up PostgreSQL for your particular scenario.

Front-End Proxy (Optional)

You can set up a front-end proxy if you'd like. This can be a simple HTTP server, something like Apache HTTPD or nginx, These can be useful for managing multiple URLs or sites through a single server machine, configuring HTTPS with SSL certificates without involving Tomcat, and so on. They are completely optional.

Configuring

XNAT 1.7 has a much lighter pre-installation configuration process than earlier versions of XNAT. You need to create a number of folders, including the folder to be used as **xnat.home**:

- Create the folder for **xnat.home**. If you create a dedicated user for Tomcat, this can be that user's home folder. If you keep the default Tomcat user, you should set the owner to that user or at last make sure the Tomcat group has full access to the folder.
- Create the following subfolders under **xnat.home**: **config**, **logs**, **plugins**, **work**.
- Create the XNAT data folders:
  - **archive**
  - **build**
  - **cache**
  - **ftp**
  - **prearchive**
- Create the XNAT configuration file **xnat-conf.properties** in the folder **${xnat.home}/config.** You can use the default version of this file as a starting point. This file is used for initializing the database connection and setting the properties for XNAT's Hibernate configuration. If you don't add the database connection and Hibernate properties, XNAT will use its own default properties, which are the same as those in the default version linked above.

This folder structure looks like this:

```
/data/xnat
        \-- archive
         |- build
         |- cache
         |- ftp
         |- home
         |    \- config
         |    |- logs
         |    |- logs
         |    |- plugins
         |    |- work
         |- pipeline
         |- prearchive
```

For XNAT development servers, we create this under **/data/xnat/home**. Because this folder needs to be writeable by the Tomcat user, it can be convenient to create the user on your system, set that user and its group as the Tomcat user and group, and then make sure all of the folders in **xnat.home** and the XNAT data folders are set to the same username and group. You can create all of these folders with the following commands on Linux platforms (this presumes that you are using **xnat** as your Tomcat/XNAT user and **/data/xnat** as the top-level folder):

Creating XNAT data folders

```
$ mkdir -p /data/xnat/home/config
$ mkdir /data/xnat/home/logs
$ mkdir /data/xnat/home/plugins
$ mkdir /data/xnat/home/work
$ mkdir /data/xnat/archive
$ mkdir /data/xnat/build
$ mkdir /data/xnat/cache
$ mkdir /data/xnat/ftp
$ mkdir /data/xnat/pipeline
$ mkdir /data/xnat/prearchive
$ chown -R xnat:xnat /data
```

Integrating the XNAT home folder with the XNAT data folders isn't necessary: the XNAT user's home folder could be the standard **/home/xnat** structure. However, it's not particularly difficult to configure a Linux server with a different home folder for a user and having all of the folders in a single structure makes for easier navigation, maintenance, and back up.

The configuration below shows the sample or default configuration for XNAT, which you can use as a starting point for your own **xnat-conf.properties** file. If you've used XNAT previously, this may look familiar. In fact, it's a snippet of the XNAT 1.6.*x* **services.properties** file. Most other properties are now set through XNAT's administration interface rather than via text files on the file system.

Sample XNAT configuration

```
datasource.driver=org.postgresql.Driver
datasource.url=jdbc:postgresql://localhost/xnat
datasource.username=xnat
datasource.password=xnat

hibernate.dialect=org.hibernate.dialect.PostgreSQL9Dialect
hibernate.hbm2ddl.auto=update
hibernate.show_sql=false
hibernate.cache.use_second_level_cache=true
hibernate.cache.use_query_cache=true
```

There are actually default values within XNAT for all of the datasource and Hibernate properties. In fact, those default values are actually the same as those shown in the sample configuration above. What this means is that you can actually start XNAT with no configuration properties specified and, as long as you've created the **xnat** database on your localhost with the appropriate username and password (or configured PostgreSQL itself to allow trusted connections), your XNAT should start up properly.

Unlike previous versions, XNAT 1.7 does not require a separate step to import core or custom data types. This removes the need to run any setup, update, or SQL scripts prior to starting XNAT. All database schema are created and initialized upon application startup.

Accordingly, there is no need to configure any files with the location of your XNAT data folders. These are also initialized on application startup. However, when XNAT first starts up and initializes, it will recognize that it is coming from an uninitialized state. Upon first logging in, you'll have a chance to review all of your configuration options and set the appropriate values.

Installing

Installing XNAT 1.7 is much simpler than previous versions. You have a number of options for how you do this, though:

- The easiest way is to simply deploy the XNAT 1.7 war file. A *war file* is a "web application archive" and contains the entire XNAT application in a single bundle. You can simply copy this file into your Tomcat **webapps** folder. If Tomcat is already running, it should automatically deploy the XNAT application. If Tomcat is not running, just restart the service and the application will be deployed automatically on startup. If you want XNAT to be the root application on your Tomcat server, you should rename it to **ROOT.war**.
- You can also "explode" the war file, which is to say, extract all of the contents into a folder underneath the Tomcat **webapps** folder. As with the war file, if you want XNAT to be the root application, you should name this folder **ROOT**.
- You can use the Tomcat manager application to deploy XNAT to your server. For this to work, you need to modify the manager configuration. Specifically, you need to modify the file **web.xml** within the manager application (if you install this via a Linux package manager, you can usually find this file at **/usr/share/tomcat7-admin/manager/WEB-INF/web.xml**). You need to change the value of the setting **<max-file-size>** to something larger than the war file, probably about 150MB. Once you've made this change, you'll need to restart Tomcat (the manager application can start and stop other applications, but not itself!).
- You can also deploy directly from the XNAT source folder using the Gradle build tool. This is not a standard practice for test or production environments, so this is covered in the XNAT Developer Guide.

Once you have installed XNAT (and restarted Tomcat if necessary), you can log in. If this is a new XNAT installation (i.e. not an upgrade of an existing XNAT site), you can use the default login username **admin** with the password **admin**.

> For security purposes, you should immediately change the password for your **admin** account! Also, if you are configuring a production deployment of XNAT, we strongly recommend that you create a new personal administrator account and disable the default **admin** account altogether.

Upon login on a new XNAT system, you'll be directed to the **Configuration** page. You should carefully review the settings on this dialog before proceeding! This configuration step is analogous to previous versions' pre-build configuration. Certain settings–specifically the folder locations for the various XNAT data folders–can be problematic to change once XNAT has been initialized.

Extending

You can extend XNAT through the use of plugins. Plugins are simply jar files that contain a mix of datatype schema, Velocity templates, JavaScript files, JSP pages, compiled Java classes, and more. To install a new plugin:

1. Stop the Tomcat service.
2. Copy the plugin jar into the **plugins** folder under your **xnat.home** folder.
3. Restart Tomcat.

That's all there is to it. Any datatype schema in the plugin will be automatically processed by XNAT and configured as a new datatype.

> In previous versions of XNAT, you had to configure any new datatypes you added to XNAT. XNAT 1.7 automatically configures new datatypes and makes them available for immediate use. You still may want to configure any new datatypes just to set the readable names and check the security settings for it, but you don't need to do that just to create new instances of your datatypes.