

What's New In XNAT

Despite remaining fairly recognizable in the front end, XNAT 1.7 has changed quite a bit under the hood. Here is a quick overview of the new features.

Version-specific Release Notes

- [XNAT 1.7.5.4-1.7.5.6 Release Notes](#)
- [XNAT 1.7.5.3 Release Notes](#)
- [XNAT 1.7.5.1 Release Notes](#)
- [XNAT 1.7.4.1 Release Notes](#)
- [XNAT 1.7.4 Release Notes](#)
- [XNAT 1.7.3.1 Release Notes](#)
- [XNAT 1.7.3 Release Notes](#)
- [XNAT 1.7.2 Release Notes](#)
- [XNAT 1.7.1 Release Notes](#)

Completely overhauled, war-only installation process

The XNAT installation process no longer requires the **xnat_builder**, and there is no need to run **setup** or **update** scripts. In place of the builder, XNAT uses a Gradle plugin to generate code from data-type schema files. The XNAT application is now delivered as a .war file, which can be dropped right into your Tomcat 7 server. There is no source code or folder to worry about.

Additionally, since the database schema generation and updates are managed on the fly, which will make it far easier to upgrade XNAT in the future.


See: [XNAT Installation Guide](#)

Also, by using Vagrant and VirtualBox, you can have a development version of XNAT running in a virtual machine. The process literally takes just a few minutes, and works on all major platforms.

See: [Running XNAT in a Vagrant Virtual Machine](#)

Consolidated configurations, easy browser-based setup and overhauled Admin UI

XNAT Site Setup

 The settings below need to be configured before this XNAT system can be used. Please set the properties below and submit the form to continue.

Site Information

Site ID

The id used to refer to this site (also used to generate ids). The Site ID must start with a letter and contain only letters, numbers and underscores. It should be a short, one-word name or acronym which describes your site.

Site URL

The address you want visible to users in emails, and other external links. This should be a functional address (i.e. if the user pasted this address in their web browser, they should come to the site). localhost only works if the web browser is located on the same machine. You are required to guarantee that this address is functional for reaching the site.

Administrator Email Address

Email address for site administrator.

Data Storage

Archive Location

Pre-archive Location

Cache Location

Build Location

In previous instances of XNAT, administrators had to wrangle a bunch of settings in properties files during the installation process. This meant that these settings were either manually duplicated or unavailable in the Administration UI once the application was built. In XNAT 1.7, we have radically reduced the amount of properties that must be set prior to installation*, and moved a core set of first-time-settings into a setup screen that displays in the UI on your first launch. Each of these settings is persisted in the database and can be revisited later via the UI as needed.

(* Exception: The PostgreSQL and Hibernate database connection strings must be set in an `xnat-conf.properties` file. But that's it, really!)

See: [XNAT Setup - First Time Configuration](#)

The Admin UI has also received a major overhaul in appearance and usability, as we attempt to lower the barriers to entry for first-time users. The rendering of the UI uses a new [XNAT Spawner](#) library for form elements, which will be more widely deployed throughout the XNAT application in subsequent releases.

Modules have been replaced by a new Plugin architecture



XNAT 1.6 Modules are not compatible with XNAT 1.7. These and other customizations will need to be refactored to comply with the XNAT 1.7 plugin structure.

Modules were introduced with some fanfare in XNAT 1.6. However, this implementation was one-way only: once a module was added to a running instance of XNAT, it could not be removed without significant effort.

XNAT 1.7 has a true plugin architecture, where extensions to XNAT are treated as separate projects by the application. Data types and new functionality can be added via plugin now, and each plugin can generate custom administration functions in the rebuilt Admin UI.

See: [Deploying Plugins in XNAT](#)

New, self-documenting REST API using Swagger

XNAT REST API

The XNAT REST API (XAPI) functions provide access to XNAT internal functions for remote clients.

Created by XNAT
 See more at <http://www.xnat.org>
[Contact the developer](#)
[Simplified 2-Clause BSD](#)

anonymize-api : XNAT DICOM Anonymization API		Show/Hide List Operations Expand Operations
GET	/xapi/anonymize/projects/{projectId}	Gets the project-specific anonymization script.
PUT	/xapi/anonymize/projects/{projectId}	Sets the project-specific anonymization script.
GET	/xapi/anonymize/projects/{projectId}/enabled	Indicates whether the project-specific anonymization script is enabled or disabled.
PUT	/xapi/anonymize/projects/{projectId}/enabled	Enables or disables the project-specific anonymization script.
GET	/xapi/anonymize/site	Gets the site-wide anonymization script.
PUT	/xapi/anonymize/site	Sets the site-wide anonymization script.
GET	/xapi/anonymize/site/enabled	Indicates whether the site-wide anonymization script is enabled or disabled.
PUT	/xapi/anonymize/site/enabled	Enables or disables the site-wide anonymization script.
automation-api : Automation Service API		Show/Hide List Operations Expand Operations
dicom-scp-api : XNAT DICOM SCP management API		Show/Hide List Operations Expand Operations

New XAPI functions have been added to XNAT in parallel to the existing REST API, which still functions as before. The benefit of the new XAPI functions is that they are discoverable and self-documenting within the Admin UI using the Swagger tool. This allows you to browse and execute XAPI commands right in your browser, which can be an immensely valuable tool during script development and testing.

Support for new features in development

Support for Automation

XNAT 1.6.5 introduced (very quietly) the ability to add scripts and automate their execution based on XNAT workflow events, using the Administration UI. In XNAT 1.7, this feature has been enhanced. The XNAT application can execute Python, Groovy or Javascript scripts that are managed in the UI. Additionally, an application-wide event framework has been introduced, allowing users to schedule tasks or have them execute via event handlers. This feature will continue to receive quite a bit of development and support in future versions.

Support for Container Services

As we demoed in the XNAT Workshop, it will be possible to link XNAT to a Docker Container and perform processing on a Docker Image. This feature is still in development, and is still in pre-beta, but is very exciting.

See: [Getting Started](#)

Support for Themes

XNAT 1.7 offers nascent support for UI theme development and deployment. This feature offers a new way to create and manage UI customizations and tailor those customizations to specific user roles.

Support for XNAT-to-XNAT project data sync

The XSync Plugin is being developed for XNAT 1.7, which will allow administrators and project owners to sync their data to a remote XNAT, essentially streamlining one of the major use cases of the [JAAT](#) tool.

What Hasn't Changed in XNAT 1.7?

The underlying architecture for 1.7 is mostly the same as 1.6:

- All of the data types are the same

- Security and user data are the same
- Most existing configuration data will migrate quietly