

NEMESIS V3 Resource Repository Guide

Date

May 21, 2015

December 5, 2016 (file naming in NemsisStates repositories)

September 26, 2023 (updated links, updated Public Git Folders list)

Authors

Joshua Legler – NEMESIS Consultant

Jorge Rojas Jr. – NEMESIS Data Manager / Analyst N.

Jennifer Correa – NEMESIS Business Data Analyst

Clay Mann – NEMESIS P.I.

Overview

This guide helps software developers integrate national and state-specific resources from the NEMESIS V3 Resource Repositories into their development environments or software products. This document provides instructions for accessing the NEMESIS V3 Resource Repositories using multiple methods.

The NEMESIS TAC uses the Git distributed version control system to provide its versioned national resources and state-submitted resources. As a version control system, Git manages revision control, change logging, and data integrity, and it supports distributed development. Knowledge of Git is assumed in order to use this guide (for further reading, see Additional Resources, p. 5).

NEMESIS stakeholders have access to the NEMESIS Resource Repositories via Git as well as via web-based and web services interfaces provided by Atlassian Stash.

Quick Start

The quickest way to access the NEMESIS V3 Resource Repositories is to visit

<https://git.nemesis.org/projects> and follow the links to the [NemsisPublic](#) and [NemsisStates](#) projects.

Repositories

NemsisPublic

NemsisPublic is the Stash project for the public repository for versioned national NEMSYS V3 resources. The repository is named `nemsis_public`. The repository is accessible via web browser at https://git.nemsis.org/projects/NEP/repos/nemsis_public/browse.












NemsisStates

NemsisStates is the Stash project for public repositories for state-specific NEMSYS V3 resources. The project includes a separate repository for each state, named using the name of the state in lowercase with hyphens between words. For example, the state repository for West Virginia is accessible via web browser at <https://git.nemsis.org/projects/NES/repos/west-virginia/browse>. The list of all state repositories is accessible via web browser at <https://git.nemsis.org/projects/NES>.

File and Folder Structures

NemsisPublic

The `nemsis_public` repository has the following folders:

 Compliance	Compliance policy documents and resource files
 DatabaseScripts	Database scripts
 DataDictionary	Data Dictionary files (HTML, PDF, text)
 DefinedLists	Defined Lists
 SampleData	XML file samples
 Schematron	National Schematron files and resources
 SuggestedLists	Suggested lists that are not subject to UMLS licensing
 Translation	V2-to-V3 translation and V3-to-V3 translations
 UsageGuides	Usage guides
 WSDL	WSDL file
 XSDs	XSD files

Generally speaking, file names do not include version numbers or dates. Version numbers, dates, and other revision history are maintained by the repository itself. A file keeps the same file name across versions over its lifetime.

NemsisStates

Each state repository may contain the following folders with documents and supporting resources. If a folder does not exist, the state has not published any resources within that topic. Certain files, as noted below, are named consistently across all states, prefixed with the postal abbreviation of the state followed by an underscore (for example, "UT_").

📁 DataDictionary	State-specific data dictionary files
📄 DataDictionary.(pdf docx ...)	Data dictionary
📁 Resources	State-specific resource list files
📄 StateDataSet.xml	StateDataSet file containing custom data elements; state required elements; state certification/licensure levels; procedures, medications, and protocols permitted by the state; EMS agencies; and facilities
📁 Schematron	State-specific Schematron rule files and resources
📄 DEMDataSet.sch	Schematron rule file for DEMDataSet
📄 EMSDataSet.sch	Schematron rule file for EMSDataSet
📁 WebServices	State-specific WSDL file and resources
📄 NEMSIS_V3.wsdl	WSDL file

Repositories may contain other files provided by states. Except for those files noted above, file names are preserved as submitted by the state, except that version numbering and revision date information are removed from file names so that resources can keep the same file names across versions over their lifetime. The repository tracks revision history and versioning. If a state utilizes version numbering, the version numbering is identified using branches or tags in the repository.

Repository Branches, Tags, and Workflow

Git is a *distributed* version control system, meaning each person who clones a repository has a full copy of that repository (including full history) on his/her computer. This section outlines the workflow and branching models utilized by the national and state-specific repositories.

NemesisPublic

The `nemesis_public` repository contains a master branch and `release-*` branches. The `release-*` branches are named for the public release of the NEMSIS V3 standard they represent. For example, NEMSIS V3.3.4 can be accessed on the `release-3.3.4` branch. Candidate releases are published on a `release-*` branch. Only final public releases are published on master. The master branch follows the most recent `release-*` branch that has final public release status.

Repository updates that contain new builds of the NEMSIS standard are tagged with the full release version and build number (for example, `3.3.4.140328`). Repository updates that affect ancillary files (compliance policy, suggested lists, usage guides, etc.) are not tagged.

NemsisStates

Each state-specific repository has a master branch. A state repository may also have release-* branches if needed to support multiple versions of the NEMESIS standard.

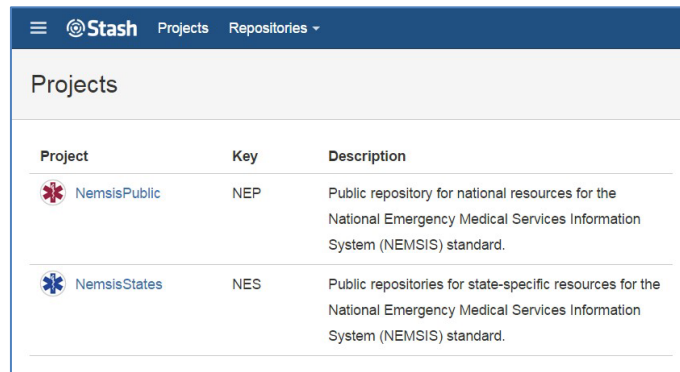
States submit updated resources to the NEMESIS TAC. NEMESIS TAC staff review and validate the submitted files before approving them for release in a state repository. States may choose to involve their vendors or IT staff in using a Git-based workflow to submit updated resources for approval and release in their state-specific repository. For more information, contact the NEMESIS TAC.

Accessing Repositories: NEMESIS Web Site



All repositories are automatically synchronized to the www.nemesis.org site. Resources in the repositories can be accessed through the traditionally available pages of the NEMESIS Web site, such as the V3 Downloads pages and the State and Territory Information pages.

Accessing Repositories: Atlassian Stash

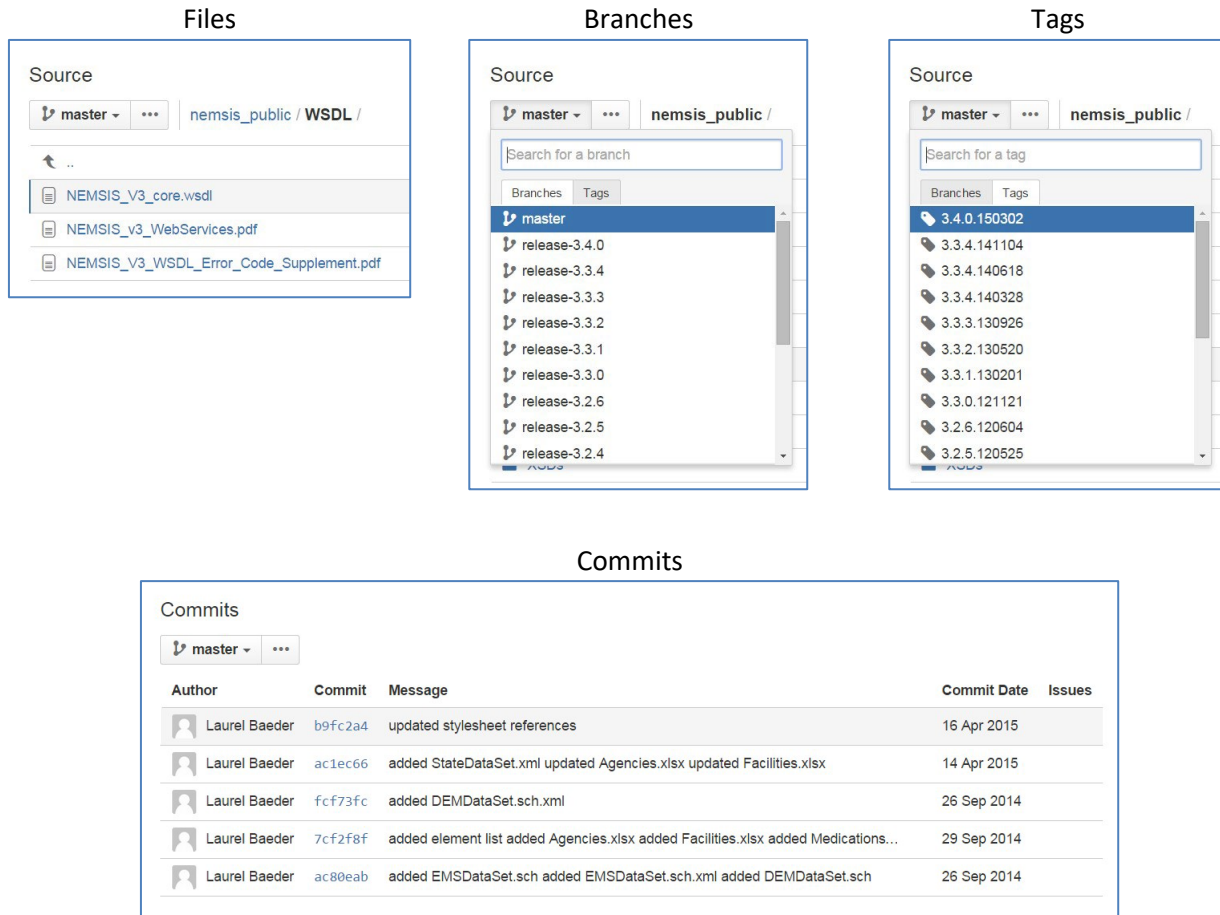
The NEMESIS TAC uses Atlassian Stash to provide a web-based interface to the NEMESIS repositories, including their contents, branches, tags, and history. The Stash server can be accessed at <https://git.nemesis.org/projects>.



The screenshot shows the Atlassian Stash web interface. The top navigation bar includes the Stash logo, a hamburger menu, and the text 'Projects' and 'Repositories'. Below the navigation bar, the page title is 'Projects'. A table lists two projects:

Project	Key	Description
 NemesisPublic	NEP	Public repository for national resources for the National Emergency Medical Services Information System (NEMESIS) standard.
 NemesisStates	NES	Public repositories for state-specific resources for the National Emergency Medical Services Information System (NEMESIS) standard.

Via Stash, you can find and download files, browse by branch or tag, and view commit history.



Accessing Repositories: Git

You can clone the NEMSIS repositories to your own computer or server using Git. The following Git command clones the nemsis_public repository into a folder called nemsis_public:

```
git clone https://stash.utahdcc.org/stash/scm/nep/nemsis_public.git nemsis_public
```

State repositories can be cloned using the name of the state in lowercase with hyphens between words. For example, the following Git command clones the West Virginia state repository into a folder called west-virginia:

```
git clone https://stash.utahdcc.org/stash/scm/nes/west-virginia.git west-virginia
```

Once you have cloned a repository, you can set up automated jobs to watch for, download, and process changes in the repository.

Accessing Repositories: REST Web Service

Atlassian Stash provides a REST Web services API for automated interaction with the NEMSIS repositories. For API documentation, visit developer.atlassian.com/cloud and select the links listed under “REST APIs.” As an example, the following REST request retrieves a list of files in the West Virginia state repository, returned by the Stash server in JSON format:

```
https://stash.utahdcc.org/stash/rest/api/1.0/projects/NES/repos/west-virginia/files
```

If a state’s repository is empty, the Stash server will return an HTTP 401 (Unauthorized) status code.

The REST API provides an easy way for software developers to build automated processes around the NEMSIS repositories.

Additional Resources

If you’re unfamiliar with Git, the following free resources may help you get started. Inclusion in this list does not imply endorsement by the NEMSIS TAC.

- Git clients:
 - [Official Git Client](#) for Windows, Mac OS X, Linux, or Solaris
 - [TortoiseGit](#) for Windows
 - [Atlassian SourceTree](#) for Windows or Mac OS X
 - Git learning materials:
 - [Git Succinctly](#) by Ryan Hodson
 - [Pro Git](#) by Scott Chacon and Ben Straub
 - [Try Git](#) by Code School
 - [Use Visual Studio with Git](#) by Microsoft

Conclusion

The Git-based NEMSIS repositories provide several advantages to software developers:

- NEMSIS resources are developed and released using a well-established workflow that provides for concurrent development, revision control, and historical consistency.
- The NEMSIS repositories follow expected software development conventions.
- NEMSIS resources use consistent folder structures, file names, and formats across national and state-specific repositories and across time.
- NEMSIS resources can be accessed via the NEMSIS Web site, Atlassian Stash web interface, REST API, and Git, all synchronized with each other.