Pukka-j

# Nexus Release Notes

Version 3.07.00

Pukka-j Limited

# Version 3.07.00

**Release Date:** June 25, 2025

## 1 Introduction

This release introduces some new features for main the Nexus application and has extended the functionality of some features from the previous release.

## 2 New Features

### 2.1 DICOMDIR Support

DICOMDIR support has been added for some backend service components. Added for DICOM upload to PACSPost.com in DICOMDIR zip format.

### 2.2 DICOM Study Zip Output Node

A new DICOM output node which monitors the output directory and zips up the study folder contents when no more images are arriving and optionally creates a DICOMDIR file to index the data. Can be used with the upload service to automatically upload DICOM studies to PACSPost.com


***Creates a zip archive of DICOM output files.***

The DICOM Zip Output node writes out DICOM files to the file system in a similar way to the DICOM File Writer node but has an extended process which monitors the output directory for new files and when no more files have arrived for a set period, the output folder is zipped up at the study level.

The configuration for the node is shown in Figure 1.

The output directory is the location where the files are written to with the subfolder name being the DICOM study UID of the images. The images are cached in a hidden folder (.cache) in the output location and the study folder is zipped up once no more images have been written into the output location with the time delay defined by the second field in the figure. A DICOMDIR index file can be optionally written as part of the output to create an index that a DICOMDIR reader can use to list the contents of the directory. The last two settings relate to the DICOMDIR, where the saving AET can be set along with the “File set ID” which is a description of the media and is used in the DICOMDIR.


***Figure 1: DICOM study zip output node configuration.***

## 2.3 PACSPost.com

A new image sharing service has been created using Nexus as the underlying engine for the image transfers. The service is an action based service whereby DICOM data can be uploaded to the service either via a web app front end or via a Nexus configured to connect with the service. Uploads have follow on actions like “share with patient” or “view”. In the case of sharing studies, users can be invited to use the service via an automatic email and studies shared using the recipient’s email. Various features have been added to Nexus support integration with the service like 2FA, QR code generation, self registration, websocket upload chunking for upload of large files.

## 2.4 Nexus Node Cleaning

A Nexus cleaning service has been added as a corollary to the existing purge service. The new clean service is intended for implementing services to create a database entry to queue such cleaning jobs for the clean service to execute. The intention is for Nexus network nodes and secondary services to be able to dictate individualised cleaning schedules and requirements. The first service to implement the new cleaning service is the *File Upload Service* (section 2.5). The will be more to come in future releases.

## 2.5 File Upload Service

The new file upload service monitors a set of configurable directories for new files that match a configurable pattern. Matching files are then queued for upload by adding an *UploadJob* entry into the configured database. The upload service has been designed specifically to work with *PacsPost* and other web-based Pukka-j portal services. Upload is authenticated by a JSON web token (JWT). It is possible to configure multiple directories to scan for files to upload and each directory can have a unique configuration for upload, such as differing URL or to/from usernames associated with *PacsPost*.

## 2.6 Nexus Error Monitoring

A new service has been added to the Nexus secondary services to provide a dedicated central error monitoring service and dashboard. Nodes have been configured to register with the service and in the event of an error, the service is notified so that the error can be displayed in a dashboard. Some errors are considered transient and therefore self clearing, such that if a connection is down, the error will be displayed but if that connection is restored, the error will be cleared from the dashboard. Some errors require a manual intervention because they may be related to a specific study or message and requires investigation.

## 2.7 Nexus Configuration History and Backup

When saving a Nexus config, the system now makes a copy of the original settings file before saving the new configuration to the current.mnx file. Access to the history is available from the Nexus user interface from option buttons at the top right of the network configuration as shown in Figure 2.


***Figure 2: Network configuration options.***

The third option opens a configuration history dialogue as shown in Figure 3.

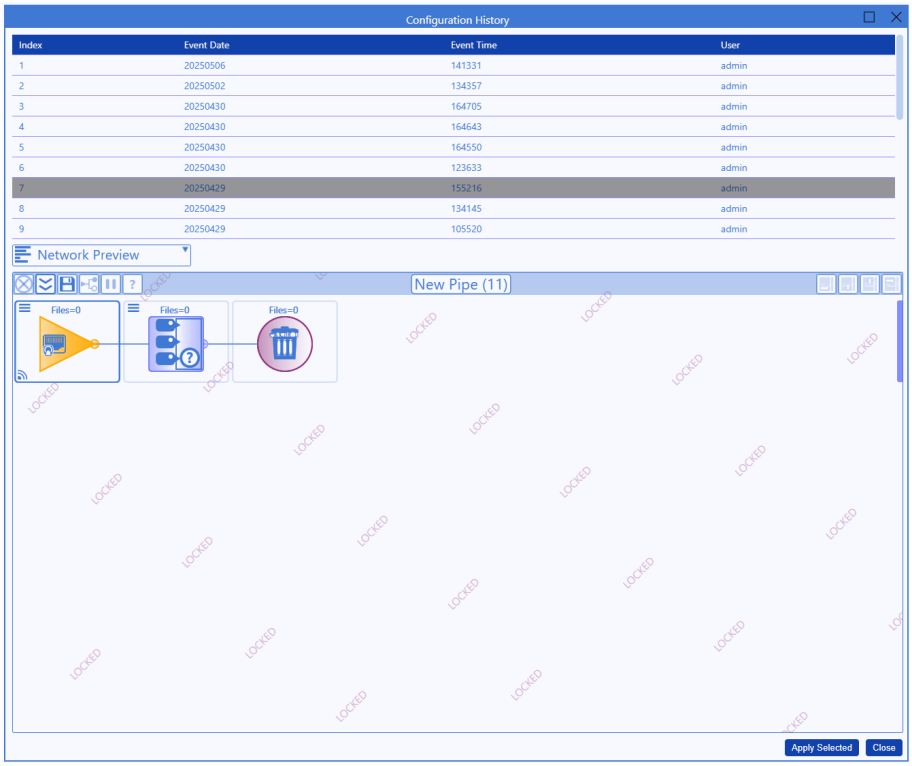



***Figure 3: Network configuration history.***

Each backed up configuration can be previewed in the user interface and can be restored by selecting the “Apply Selected” option at the bottom right of the dialogue.

## 2.8 Nexus Node and Pipe Configuration Saving

It is now possible to export the individual node configuration and individual pipe configuration from within the Nexus user interface.

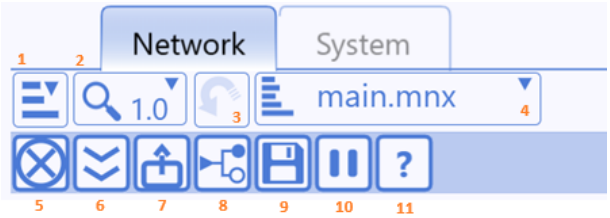



***Figure 4: Network controls.***

Figure 4 shows the primary controls of the network (top row) and the controls available on an individual pipeline where option seven is to save the pipe configuration. Selecting that option will display the serialised configuration of the pipeline which can be exported to the downloads folder by pressing the save icon at the top right. The download takes the name of the pipeline when it is saved to the downloads. A previously saved pipeline can be added to a network by dragging and dropping the file onto an unlocked Nexus network.

Individual node configuration can be exported from within the individual node configuration dialogue. Each node now has a “node config” tab where the name of node can be defined and a text field at the bottom where an output filename can defined. When the “Export Config” button is selected, a filename with the name defined in the text area is saved to the downloads folder. Previously saved node configurations can be applied to new nodes simply by dragging the exported file onto the node configuration dialogue.

It is now possible to add or change indivual pipelines and run them without having to interrupt the entire network. Using item 9 in the figure allows users to apply that specific pipeline configuration to the server without having to select the option to apply the entire network.

## 2.9 Nexus HL7 Caching Output Node

A new status view on the HL7 caching output node has been added to give feedback and control on the output queue.

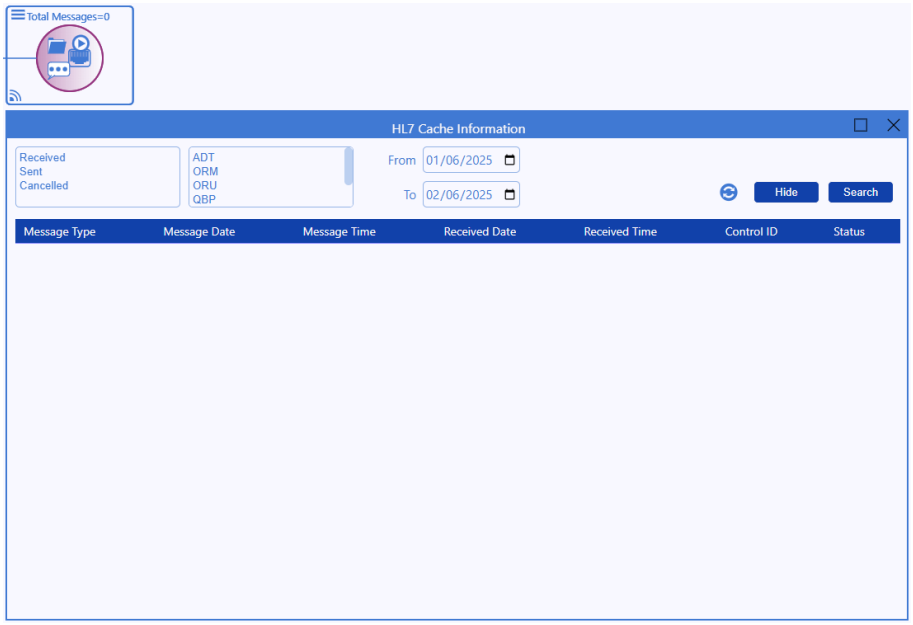



***Figure 5: HL7 Caching Ouput Node Status View.***

Figure 5 shows the status view available from the node. The messages can be filtered by status, type and received date. The messages can be reset or cancelled from the queue by right-clciking on messages listed in the query results table.

# 3 Improvements

## 3.1 2FA

Added Base32 support for two factor authentication.

## 3.2 Magic Byte Detection

Imporoved the magic byte logic for file type detection when reading input streams.

## 3.3 Improved Logging

General logging improvements and stack trace output.

## 3.4 Extended JWT Support

Improved the JWT usage for integration with other systems and also cross launching of Nexus services.

## 3.5 Extended Java DB Support

Better coverage for support of Java DB in the nodes and services has been added.

## 3.6 Applications Homepage

The apps.html homepage has been improved and updated to give a better experience for launching the extended apps within the Nexus platform.

The launchable apps are controlled using the apps.json config file.

```
{install_location}/docs/config/server/apps/apps.json
```

The apps.json file defines which application entry points are available to users when the apps.html page is used.


***Figure 6: Pukka-j Application Launcher Homepage.***

Figure 6 shows the homepage served out by the apps.html file. The drop down options are defined by the apps.json file. Selecting an option and pressing the launch button will open the selected app on a new tab. The example shows the PACS Manager app being selected and the tab which appears when launched.

The **PACS Manager** application allows users to manage stored DICOM image data and their associated databases.

The list of available databases is shown in the top-left panel of the page. These databases correspond to the database nodes created within **Nexus** and are displayed here for management purposes. When creating a database node in Nexus, a node name can be defined in its configuration, which is then shown in this list, as illustrated in Figures 7 and 8.

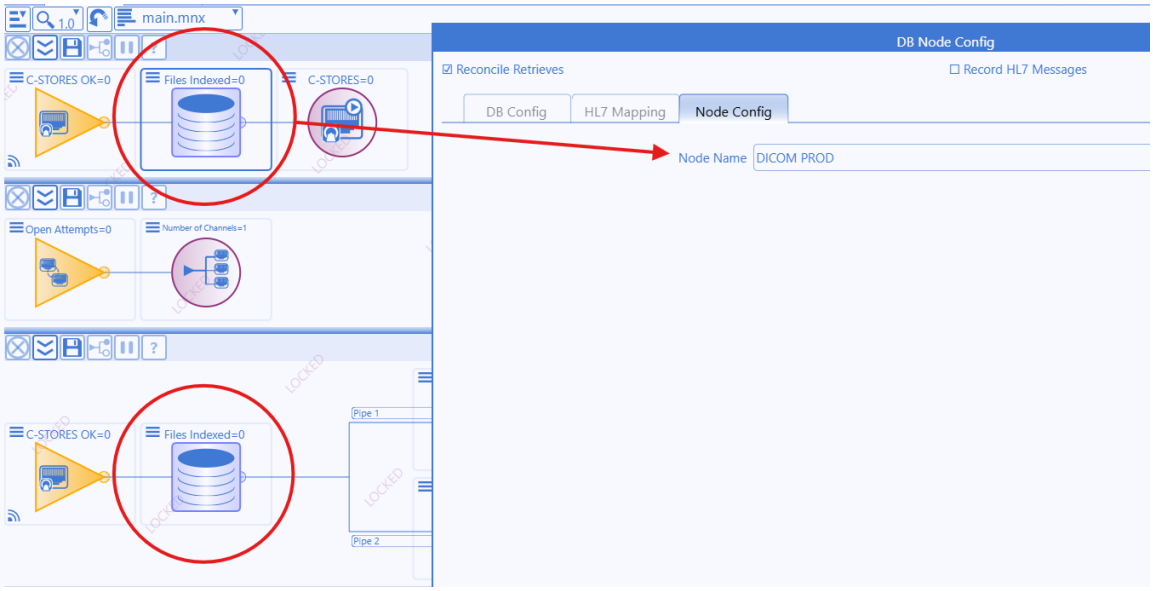



***Figure 7: Nexus database node and its configured node name.***

Once a database is selected, you can query it using the query bar, with options such as date range, modality, and more. An advanced search is also available for filtering by Study ID, Study UID, or Procedure Code. Figure 8 shows the query interface of the PACS Manager application.

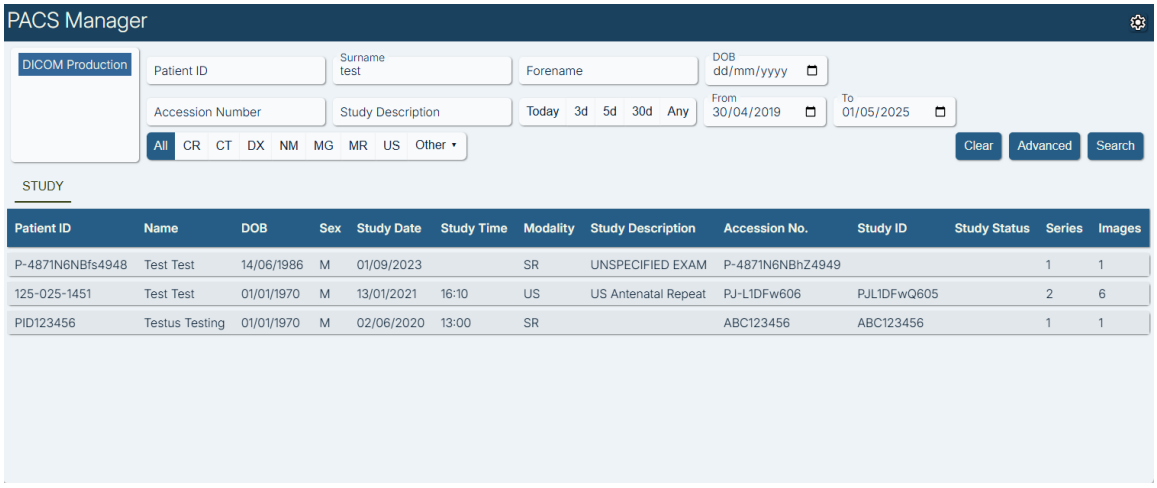



***Figure 8: PACS Manager database query interface.***

### Navigation

Once a search is performed on a selected database, the **Study Table** tab appears, displaying all studies that match the criteria. Each database includes two additional tabs: the **Series Table** and the **Image Table**. Each table provides its own set of modification options, which are discussed in Section 3.6.

To navigate:

- Select a study in the Study Table to access its corresponding Series Table.
- From the Series Table, select a series to access the Image Table.
- Alternatively, double-clicking a study or series will open the respective next level directly.

Figures 9 and 10 illustrate the Study and Series Table views.

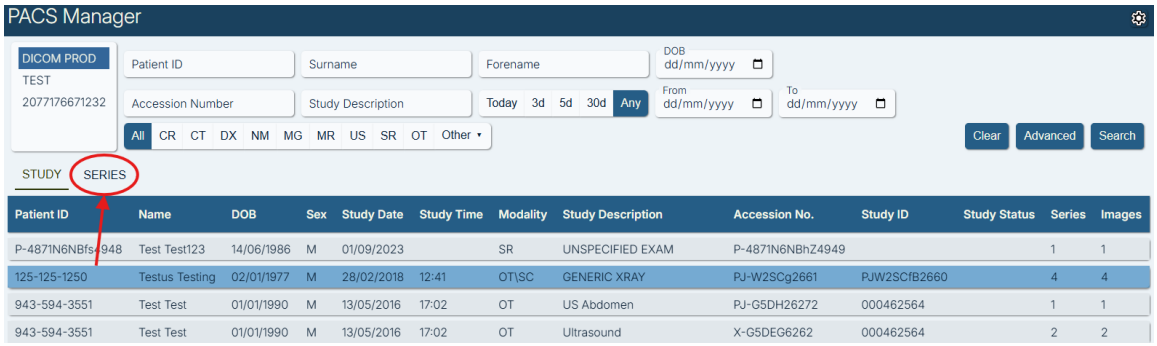



***Figure 9: Study Table view with a selected study.***

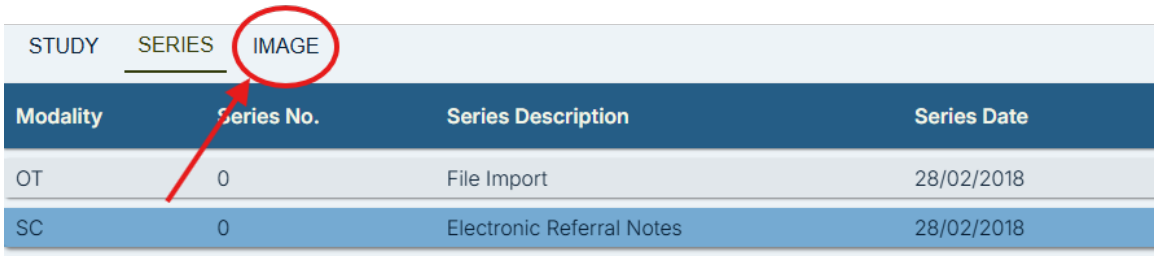



***Figure 10: Series Table view with a selected series.***

### Options

Each table supports right-click context menus with various options depending on the selection:

**Multi-selection** When multiple rows are selected in any table, the only available action is to delete them, as shown in Figure 11. Once confirmed, this will remove the entries from the database and delete their associated files (if stored).

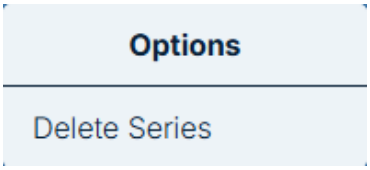



***Figure 11: Delete option for multiple row selection.***

**Study Table Options** When a single study is selected, the following options are available:

- **Edit patient details**: Opens a dialog to modify patient information. An option is provided to update the image files themselves with the new details.
- **Edit study**: Allows editing of study-level metadata with an optional update to the underlying files.
- **Set study status**: Change the status of a study by selecting applicable statuses. Status indicators are displayed in the Study Table.
- **Delete study**: Prompts a confirmation dialog to permanently remove the study.
- **Open audit view**: Displays a log of all actions performed on the selected study.

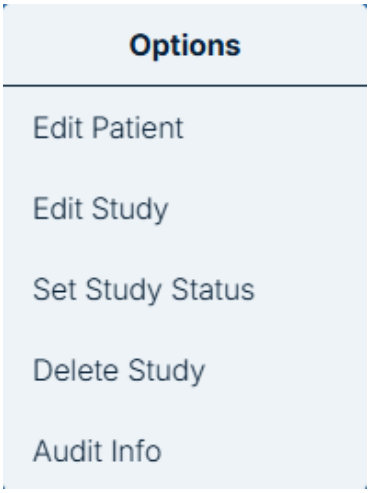
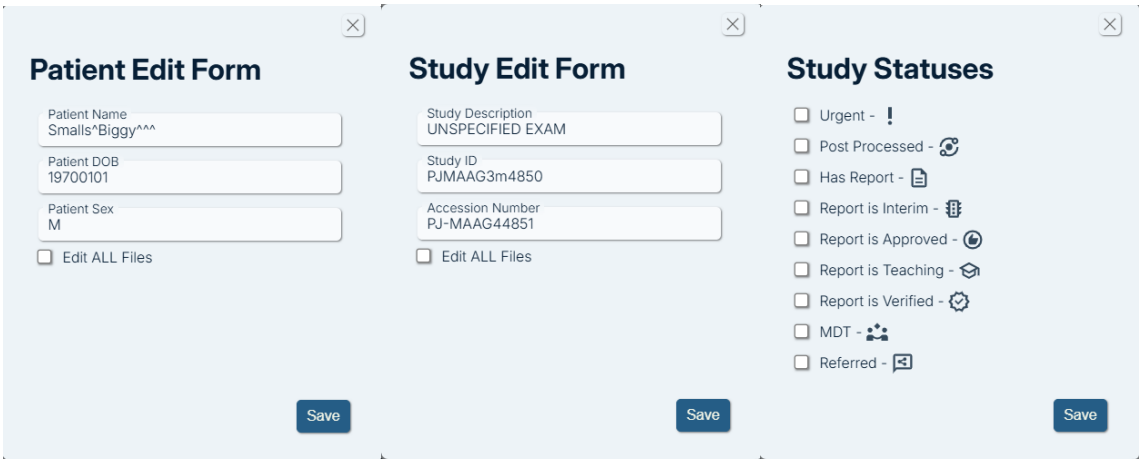



***Figure 12: Study options context menu.***

Edit dialogs are illustrated in Figure 13.

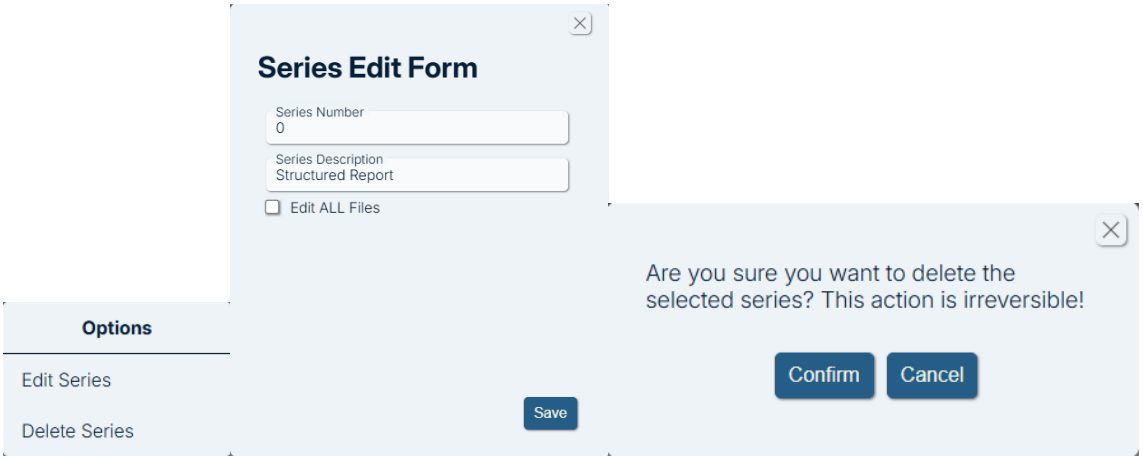



*(a) Edit Patient Dialog* *(b) Edit Study Dialog* *(c) Study Status Dialog*

***Figure 13: Edit and status dialogs for a selected study.***

**Series Table Options** The Series Table provides a subset of the Study Table options:

- **Edit series**: Allows modification of series-level details, with an option to update the files.
- **Delete series**: Prompts a confirmation dialog to remove the selected series and associated files.




*(a) Series Options Context Menu* *(b) Edit Series Dialog* *(c) Delete Confirmation Dialog*

***Figure 14: Options and dialogs for a selected series.***

**Image Table Options** In the Image Table, the only available action is to delete one or more selected images. This action also requires confirmation.

## 3.7 Patient Level Reconciliation

The data reconciler node has been updated to also include a patient level reconciliation. Up to now, the tool relied on knowledge of a DICOM study UID in order to fix up and study level. The option to update the patient details for new studies where the patient ID is already known has been added. The patient details can be learned by sending ADT messages into the database used for the reconciliation.

## 3.8 Age Node

The age node in the chip config was improved to allow HL7 message filtering based on calculated age between “study date” an “date of birth”. By default the study date is taken from ORC.0.7.0.3.0 and the date of birth is from PID.0.7.0.0.0.

## 3.9 HL7 Caching NACK Blocking

The HL7 Caching Output Node has been updated to have an option decide if a NACK from the receiver should block the queue or not. Previously, the node would block on a NACK but that would stop good messages from going over. If the blocking is turned off, the NACK will trigger an error which will be visible in the Nexus error manager to be manually cleared. Connection failures will always block.

## 3.10 Application Login Timeout

An inactivity timeout has been added to the worklflow controller application. By default, the timeout is 5 mintues. The timeout can be set by setting the timeout in the property file.

```
''inactivityMins'' : 5
```

in

```
install_location/docs/config/server/workflow-manager.json
```

## 3.11 Workflow Service - Service Line Output

The workflow service has been updated to allow the configuration of whether or not the service line is sent in the progress messages. A new additional message can now be sent which will be the initial RQ message with the service line. This new process is to provide better feedback to external systems when inbound requests may change the service line of the job. In the case where an initial message set the service line, if a second message is used to change the service line, a “ping pong” effect could occur if the first job was in progress when the second message to change the service line arrived. By only sending the service line on the original RQ message and not on the status / progress updates, the system has better control of notifying downstream systems of the changes.

### 3.12 Workflow Service - Error Email Alert

The workflow service has been updated to allow the configuration of an email alert to notify an account if a workflow job fails.

### 3.13 Multi-frame Splitting

The NM modality has been added to the allowed modalities for the multi-frame splitter node to allow series like MIPS that have been saved as multi-frame to be split into individual images.

### 3.14 Lite Viewer Update

The lite viewer application has been updated to use a new query page and the view has been extended to be able to display DICOM SRs.

### 3.15 Nexus Redraw Efficiency

The redraw process for updating the Nexus UI on status updates has been improved to reduce that amount for redraws performed by the client UI in order to make the UI more responsive.

### 3.16 Anonymisation Node DICOM Tag Lookup

The lookup for the DICOM tag on the Dicom Anonymisation Node has been improved along with the UI layout.

## 4 Bug Fixes

### 4.1 HTML Character Encoding

**Issue:** The back tick character was incorrectly encoded in the String utility class.
**Resolution:** Corrected %60 (back-tick) added back-stroke html encoding.

### 4.2 HL7 Charset Parsing

**Issue:** Null Pointer on some HL7 file streams.
**Resolution:** Fixed charset issue with hidden char on end of charset value (ôn end of segment not trimmed).

### 4.3 DB Last Insert ID

**Issue:** SQL to get the last insert ID is different for different DB types but was always using MySQL version.
**Resolution:** Updated the SQL for the Java DB and MS SQL code for get the last insert ID.

### 4.4 Alert Reset

**Issue:** When an alert was reported in the Nexus UI from the server side, reset the alerts was only temporary and the alert would reappear on the next refresh.
**Resolution:** Resetting the alerts now clears the alert status permanantly.

## 5 Known Issues

None

## 6 Installation Instructions

Before making any changes, the **docs** and **lib** folders in the install directory should be backed up along with the **nexus.jar** in the install root.

The libraries required for this release are:

| | |
|---|---|
| **nexus.jar** | 3.07.00 |
| **pjk.jar** | 1.21.00 |
| **pacsology.jar** | 1.26.00 |
| **mint.jar** | 1.29.00 |

Plugin library versions are:

| | |
|---|---|
| **anon.jar** | 1.13.00 |
| **broker.jar** | 1.13.00 |
| **redact.jar** | 1.12.00 |

For Origin users:

| | |
|---|---|
| **websocket.jar** | 1.14.00 |

An update MySQL library is potentially required depending upon the MySQL service.

The UI is updated via the files:
**cloe.js.gz**
**pj.css**