

Laurel Bridge Compass User Manual



Laurel Bridge Software, Inc.
302-453-0222
www.laurelbridge.com

Document Version: 3.5.1
Document Number: LBDC-000062-030501
Last Saved: 5/24/2023 4:44:00 PM

Table of Contents

1	Compass Basics.....	1
1.1	Overview	1
1.2	Additional Reading.....	2
2	Installation.....	3
2.1	Upgrades.....	3
2.1.1	Important Note on Software Updates.....	3
2.1.2	Upgrading from a Previous Version.....	3
2.1.3	Upgrading to Version 2.2.x (or higher) from a 2.1.x Version.....	4
2.1.4	Upgrading a Failover Cluster to 2.2.x (or higher) from a 2.1.x Version.....	4
2.2	Minimum System Specification.....	5
2.3	Prerequisites.....	6
2.3.1	Installing SQL Server 2012 Express x64 with Tools.....	6
2.3.2	Installing SQL Server 2014 Express x64 with Tools.....	7
2.4	Installation: Compass Application.....	7
2.5	Configuring Compass Database Connectivity.....	12
2.6	Configuring Compass Web Connectivity.....	12
3	Getting Started.....	13
3.1	Overview.....	13
3.2	Installing a License.....	14
3.2.1	Installing a license file.....	14
3.2.2	Activating a license over the internet.....	15
3.2.3	Activating a license that added the HL7 Edition.....	16
4	Configuration for Routing DICOM.....	16
4.1	Creating a Source.....	16
4.1.1	DICOM Source.....	16
4.1.1.1	Settings.....	17
4.1.2	Hot Folder Source.....	18
4.1.2.1	Settings.....	19
4.1.2.2	Zip File Ingestion.....	20
4.1.3	DICOMweb Source.....	20
4.1.3.1	Settings.....	21
4.1.4	Advanced Settings.....	21
4.1.4.1	Filters for Sources.....	21
4.1.4.2	General Settings.....	22
4.1.4.3	Log Settings.....	22
4.1.4.4	Logging.....	22
4.1.4.5	Outbound IP Configuration.....	22
4.2	Creating a Destination.....	23
4.2.1	Compass Settings.....	23

- 4.2.2 Destination Settings 23
- 4.2.3 Transfer Syntax Settings 24
- 4.2.4 Jobs Settings 24
- 4.2.5 Advanced Settings 26
 - 4.2.5.1 Filters for Destinations 26
 - 4.2.5.2 General Settings 26
 - 4.2.5.3 Codec Settings 26
 - 4.2.5.4 Schedule 26
 - 4.2.5.5 Log Settings 27
 - 4.2.5.6 Logging 27
 - 4.2.5.7 Transport 27
 - 4.2.5.8 DIMSE Relayer 29
 - 4.2.5.9 Use TLS 30
- 4.3 Creating a Rule 31
 - 4.3.1 Rule Conditions 31
 - 4.3.2 Rule Actions 33
 - 4.3.3 Rule Options 33
- 4.4 Special Considerations for Routing WADO-RS (DICOMweb) 34
 - 4.4.1 Mapping WADO-RS to C-MOVE 35
 - 4.4.2 Mapping C-MOVE to WADO-RS 35
 - 4.4.3 Mapping WADO-RS Through Compass Back to WADO-RS 36
- 4.5 Configuring a Job Action 37
 - 4.5.1 Execute Job Action 37
 - 4.5.2 HL7 Send Job Action 37
 - 4.5.2.1 HL7 Message 38
 - 4.5.2.2 HL7 ACK 40
 - 4.5.2.3 Where to look for demographics 40
 - 4.5.2.4 On HL7 Message Send Success 40
 - 4.5.2.5 Logging 41
 - 4.5.3 Load Balancing Job Action 41
 - 4.5.4 PowerScribe Custom Fields Job Action 42
 - 4.5.4.1 Sample mapping file 43
 - 4.5.4.2 Post Process API 44
 - 4.5.4.3 Additional Tools: PowerTools 44
 - 4.5.5 Fluency Job Action 44
 - 4.5.5.1 HL7 Message 45
 - 4.5.5.2 HL7 ACK 46
 - 4.5.5.3 Where to look for demographics 46
 - 4.5.5.4 Sample mapping file 47
 - 4.5.5.5 Post Process API 47
 - 4.5.5.6 Additional Tools: PowerTools 48
 - 4.5.6 Allscripts Job Action 48

- 4.5.6.1 Configuration 48
- 4.5.6.2 HL7 Message 48
- 4.5.6.3 HL7 ACK..... 48
- 4.5.6.4 Where to look for demographics 49
- 4.5.6.5 On HL7 Message Send Success 49
- 4.5.6.6 Logging 49
- 4.6 Creating a Listener 50
 - 4.6.1 Listener Settings..... 50
 - 4.6.2 Encryption Settings 51
- 4.7 Filters..... 52
 - 4.7.1 Conditions 52
 - 4.7.2 Actions..... 53
 - 4.7.3 Original Attribute Sequence..... 56
 - 4.7.4 Watermark 56
 - 4.7.4.1 Options..... 56
 - 4.7.4.2 Automatic Watermark Text Placement 57
 - 4.7.4.3 Manual Watermark Text Placement 58
 - 4.7.5 De-Identify Filter Action 59
 - 4.7.5.1 DICOM Attribute Tags to Anonymize 59
 - 4.7.5.2 General Settings 60
 - 4.7.5.3 Pixel Regions to Anonymize 61
 - Percentage 62
 - Negative Pixel Region Coordinates 62
 - Absolute Coordinates 63
 - Demo Button..... 63
 - Color Button..... 64
 - 4.7.5.4 One-Way Anonymization 64
 - 4.7.5.5 Modify Dates and Times 65
 - 4.7.5.6 Additional Notes 68
 - 4.7.6 Re-Identify Filter Action 68
 - 4.7.6.1 Additional Notes 69
 - 4.7.7 Composer Action Examples 69
 - 4.7.7.1 Working with DICOM sequences 71
 - 4.7.7.2 Modifying private tags 71
- 5 Routing of HL7 Messages 73
 - 5.1 Creating an HL7 Source 73
 - 5.1.1 HL7 Network Source 73
 - 5.1.1.1 Settings..... 73
 - 5.1.2 HL7 Web Source..... 75
 - 5.1.3 HL7 Hot Folder Source 75
 - 5.1.3.1 Settings..... 76
 - 5.1.3.2 Zip File Ingestion 76

- 5.2 Creating an HL7 Destination 77
 - 5.2.1 Compass Settings 77
 - 5.2.2 Destination Settings 77
 - 5.2.3 Advanced Settings..... 78
 - 5.2.4 Jobs Settings..... 78
- 5.3 Creating an HL7 Rule..... 80
 - 5.3.1 Conditions 80
 - 5.3.2 Rule Actions 83
 - 5.3.3 Rule Options..... 83
- 5.4 Configuring an HL7 Job Action 85
 - 5.4.1 Trigger 85
 - 5.4.2 Destinations 85
 - 5.4.3 Type..... 85
 - 5.4.4 HL7 PowerScribe Custom Fields..... 85
 - 5.4.5 HL7 Report To DICOM SR..... 88
- 6 System Settings 89
 - 6.1 DICOM System Settings 89
 - 6.1.1 Compass Application Logging 89
 - 6.1.1.1 Log Settings 90
 - 6.1.2 DICOM Incoming 90
 - 6.1.3 DICOM Outgoing 91
 - 6.1.4 Compass Data Storage 91
 - 6.1.4.1 Cache Configuration..... 91
 - 6.1.5 Destination Heartbeat Sensing 92
 - 6.1.6 Compass Title 92
 - 6.1.7 Custom Table Columns 92
 - 6.1.8 Compass Web Interface 93
 - 6.1.8.1 Secure Web Access (HTTPS)..... 93
 - 6.1.9 Compass TLS Certificate Configuration 94
 - 6.1.10 Compass Administrative Functions..... 94
 - 6.1.10.1 Limit Configuration Backups 94
 - 6.1.10.2 Web User Administration..... 95
 - 6.1.10.3 Compass Audit Logging 96
 - 6.1.10.4 Compass Security 97
 - 6.1.10.5 Lighthouse Configuration..... 98
 - 6.2 HL7 System Settings..... 98
 - 6.2.1 Compass Application Logging 98
 - 6.2.2 HL7 Incoming 98
 - 6.2.3 HL7 Outgoing..... 99
 - 6.2.4 Compass Data Storage 99
 - 6.2.5 HL7 Destination Heartbeat Sensing 99
 - 6.2.6 Compass Title 100

6.2.7	Custom Table Columns	100
6.2.8	Compass Web Interface / HL7 Messaging via Web	100
6.2.9	Compass TLS Certificate Configuration	102
6.2.10	Compass Administrative Functions.....	103
6.2.10.1	Limit Configuration Backups	103
6.2.10.2	Web User Administration.....	103
6.2.10.3	Compass Audit Logging	103
6.2.10.4	Compass Security	103
6.2.10.5	Lighthouse Configuration.....	103
7	Notifications	104
8	Enabling Input and Output	106
8.1	DICOM Input and Output.....	106
8.2	HL7 Input and Output	106
9	Compass Cache.....	107
9.1	Adding to the Cache and the Penalty Box.....	107
9.2	Cache Contents and Duplicates	108
9.3	Purgers	108
9.4	Cache Query/Retrieve.....	109
9.5	Cache Operations.....	109
10	Thick Client: Compass User Interface Details	111
10.1	Menu Bar.....	111
10.2	Tool Bar	112
10.3	DICOM Jobs	112
10.3.1	View Logs	114
10.4	HL7 Jobs	115
10.5	Active Associations.....	115
10.6	HL7 Connections and Nodes	115
10.7	Status Bar	116
11	Web Client: Compass Web Interface Details	117
11.1	Dashboard.....	117
11.2	Login.....	117
11.3	DICOM/HL7 Jobs	118
11.4	DICOM/HL7 Sources.....	119
11.5	DICOM HL7 Destinations.....	119
11.6	Inbound Association History	120
11.7	Cache Patients View.....	120
11.8	Cache Studies View	121
11.9	Penalty Box	121
11.10	Order Entry.....	122
11.11	Users	124
12	Additional Security Considerations.....	126

12.1	Network Connections	126
12.2	Database Connections	126
12.3	At-Rest Encryption	126
13	Custom Code Extensions.....	127
13.1	DICOM Custom Filters.....	127
13.1.1	Custom Filter Action Interface	127
13.1.2	Custom Filter Action Example.....	127
13.2	Custom Rule Conditions.....	128
13.2.1	DICOM Custom Rule Condition Interfaces.....	128
13.2.2	DICOM Custom Rule Condition Examples.....	128
13.2.3	HL7 Custom Rule Condition Interfaces	129
13.2.4	HL7 Custom Rule Condition Example.....	129
13.3	Custom Job Actions.....	130
13.3.1	Custom Job Action Interfaces	130
13.3.2	Custom Job Action Example.....	130
13.3.3	Custom HL7 Job Action Interfaces	131
13.3.4	Custom HL7 Job Action Example.....	131
13.4	Custom HL7 Rule Match Actions.....	133
13.4.1	Custom HL7 Rule Match Action Interface.....	133
13.4.2	Custom HL7 Rule Match Action Example.....	133
13.5	Custom Disk Folder Serializers	134
13.5.1	Custom Disk Folder Serializer Interfaces	134
13.5.2	Custom Disk Folder Serializer Example	134
13.6	Custom File Batch Ingestor	136
13.6.1	Custom File Batch Ingestor Interfaces	136
13.6.2	Custom File Batch Ingestor Example.....	136
Appendix A: Compass Privacy and Security Statement.....		138
1	Management of Private Data	138
1.1	Types of PHI Maintained.....	138
1.2	Persistence of Private Data	139
1.3	Transmission of Private Data	139
1.4	Payment Card Industry (PCI) Data Security Standard	139
2	Security Capabilities	140
2.1	Automatic Logoff.....	140
2.2	Audit Controls	140
2.3	User Authorization.....	141
2.4	Security Configuration	141
2.5	Security Updates.....	142
2.6	De-Identification of PHI.....	142
2.7	Backup and Restore	142
2.8	Emergency Access.....	142

- 2.9 Data Integrity and Authenticity 142
- 2.10 Malware Protection 143
- 2.11 Node Authentication..... 143
- 2.12 Person Authentication 143
 - 2.12.1 Local Web User Administration 143
 - 2.12.2 Single Sign-On (LDAP/AD) Web User Administration 143
- 2.13 Physical Locks..... 144
- 2.14 Device Life Cycle Roadmap 144
- 2.15 System and Application Hardening..... 144
- 2.16 Security Guidance 145
- 2.17 Data Storage Confidentiality 145
- 2.18 Data Transmission Confidentiality 145
- 2.19 Data Transmission Integrity 145
- 2.20 Other Security Considerations 145
- 3 GDPR Notes 147
- Appendix B: Compass FAQs..... 148
 - 1 How can a SOP class be added to the default list? 148
 - 2 How can I distinguish between my non-DICOM jobs in the user interface? 148
 - 3 How do I remove private tags from a DICOM data set? 148
- Appendix C: Communicating Securely with Compass 150
 - 1 Secure DICOM and HL7 Communication with Compass 150
 - 1.1 Overview 150
 - 1.2 Configuring Secure DICOM Communication..... 150
 - 1.3 Configuring Secure HL7 Communication 151
 - 2 Secure Communication with Compass Web 152
 - 2.1 Disabling SSL 3.0 Support..... 153
 - 2.2 Disabling TLS 1.0 Support..... 154
 - 2.3 Enabling or Disabling TLS 1.1 Support and Enabling TLS 1.2 Support..... 154
 - 2.4 Disabling Support for the RC4 Cipher Suite 155
 - 2.5 Disabling Support for the Triple DES (3DES) Cipher Suite..... 156
 - 2.6 Enabling or Disabling TLS 1.3 Support 156
 - 3 A Note About FIPS 140-2 Compliance 157
- Appendix D: Compass Configuration Backup Files 158
- Appendix E: Create and Export a Self-Signed TLS Certificate..... 159
 - 1 Using IIS Manager 159
 - 2 Using PowerShell 159
- Appendix F: Hot Folder and HL7 Hot Folder Batch Behavior 160
 - 1 Hot Folder Basics / Definitions 160
 - 2 New Configuration Fields 160
 - 3 Hot Folder Behavior..... 161

4	DICOM Hot Folder Examples	161
	Appendix G: DICOMweb to Google Cloud Platform (GCP)	165
1	Google's Cloud Healthcare API	165
2	GCP Web Console	165
2.1	Authentication Method for GCP Service Account	167
3	DICOMweb Destination	168
3.1	DICOMweb Configuration	169
3.1.1	Authorization	169
3.1.1.1	P12 Certificate	169
3.1.1.2	Json Key File	170
3.1.2	STOW-RS, QIDO-RS, and WADO-RS with GCP	170
	Appendix H: Compass Service Crash: Restart Behavior and Logging	172
1	Restart Behavior	172
2	Logging	172

1 Compass Basics

1.1 Overview

Compass acts as a router for DICOM images/messages and HL7 version 2.x messages. Compass is able to route from one or more Sources to one or more Destinations. This routing process can be described in phases as follows.

For DICOM images/messages:

- First, a Source (or DICOM SCU) initiates communication with Compass via a DICOM Association. Compass accepts this connection and receives DICOM images/messages from the Source, filtering the incoming information based upon any user-defined Filters for incoming messages.
- Next, Compass compares the image/message information against any user defined Rules. There are several different modes in which these Rules operate:
 - **Store and Forward:** in this mode, Compass must be specifically receiving a C-STORE request message. Compass schedules the received image to be sent to the Destination(s), or SCP(s), defined by any Rules that apply. Finally, the image is filtered and sent to the determined Destination(s) at the scheduled time(s). Sources, Destinations, Rules, and Filters for DICOM traffic are discussed in further detail in Sections 4.1, 4.2, 4.3, and 4.5. Additionally, the applied Rules may specify that the received image be kept in the Compass Cache, a collection of images which is organized in the traditional Patient/Study/Series/Image hierarchy. The Cache can be searched using the web interface and queried via standard DICOM C-FIND mechanisms. The Cache is discussed in more detail in Section 9.
 - **Direct:** in this mode, Compass may be receiving any DICOM message (including C-STORE request). Compass will determine a single destination to which this message should be forwarded and will set up a real-time connection to that destination, immediately forwarding the message to it, and sending that destination's response(s) back to the originator, also in real time. Again, the message will be filtered according to the Destination's Filter settings when it is forwarded, as will the response(s) from the Destination. In this mode, Compass does not remember or store any messages it receives or sends, so the responsibility for retrying or resending lies with the Source itself.

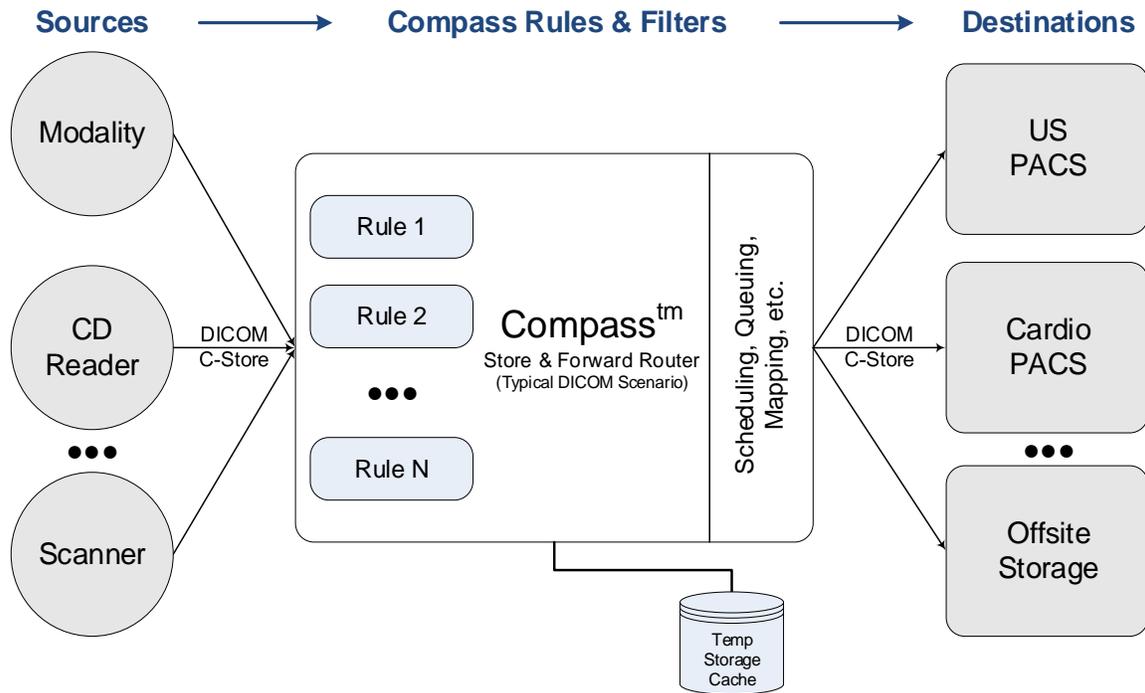
For HL7 messaging, the process is conceptually the same, with only Store and Forward supported:

- First, a Source (an HL7 sender) initiates communication with Compass by connecting to a specific listening port and sending an HL7 message.
- Compass receives the message from the source and compares the message against any user-defined Rules, scheduling the message to be sent to the Destinations(s) (i.e. HL7 receivers) referenced by any Rules that apply. Rules for HL7 messaging only have Store and Forward semantics – Direct routing does not apply.
- Finally, the message is sent to the determined Destination(s) at the scheduled time(s). Sources, Destinations, and Rules for HL7 traffic are discussed in further detail in

Sections 5.1 Creating an HL7 Source, 5.2. Creating an HL7 Destination, and 5.3 Creating an HL7 Rule.

Compass is comprised of three parts:

- The **Compass Service**, which runs as a Windows Service.
- The **Compass Client** user interface (UI), which is used to configure the Compass Service and monitor the jobs of which Compass is aware. The Compass GUI does not need to be running in order for the Compass service to run, accept, process, and send jobs.
- The **Compass Web Server**, which provides users browser access to monitor and track Compass jobs.



1.2 Additional Reading

In addition to this user manual, the following documents should be read:

- [Compass-Dicom-Conformance-Statement.pdf](#)
- [CompassDicomWebConformanceStatement.pdf](#)
- [DicomAnonymizationConformanceStatement.pdf](#)

The documents can be found in Compass' installation folder and the Laurel Bridge website.

2 Installation

2.1 Upgrades

2.1.1 Important Note on Software Updates

For running this application, we recommend that it be installed on a supported operating system and that there be a regular application of updates and security patches to that system.

Regular system backups are encouraged. A backup, especially of the application configuration data, including rules, scripts, and filters, should be made before applying any system updates. It may be “easy” to re-install the application, but it may not be easy to re-create your local configuration without a backup.

We also recommend that automatic updates be disabled on systems; while we encourage updates, especially security updates, we do recommend testing and manual application of such updates. A system administrator should manage and be present for the application of any upgrades and for any system re-boot – for whatever reason. Be wary of unintended consequences like privileges, permissions, or firewalls that change as a side-effect of patches.

Handle these activities in a controlled and planned manner; always have a plan and methodology that will allow you to back out of changes. In the event that an update proves undesirable for any reason, the process should allow the changes to be rolled back to the previous state. Most of the time things will go well, but remember that there is always the possibility that bad things will happen when you make changes.

Your operating system vendor has likely published best practices for managing patches and updates. Take the time to read them as well as to read the documentation that may be provided with any patches or updates.

2.1.2 Upgrading from a Previous Version

Prior to upgrading, make sure the license tied to the copy of Compass being upgraded is covered under a valid maintenance contract that isn't expired; licenses that don't have a valid maintenance contract cannot be upgraded.

An older Compass version can be upgraded to a newer Compass version without uninstalling the older version (unless explicitly noted as being necessary for particular cases described in the following sections of this chapter).

When upgrading a copy of Compass that is multiple versions newer than the old version, it is not necessary to install the intermediate versions; the new version will apply all the changes that occurred between the old version and the version currently being installed.

Prior to installing the new version, exit the Compass client program, and stop the Compass service.

After installing and launching the new version, Compass' About dialog may be displayed and indicate a product version mismatch (depending on the installed license type). You should activate or install the new license version as described in chapter 3 of this user manual.

2.1.3 Upgrading to Version 2.2.x (or higher) from a 2.1.x Version

Starting with version 2.2.x, Compass is a 64-bit application. It must be installed on a 64-bit version of a Windows operating system. If you are upgrading from a 2.1.x version, you must stop the Compass Service and uninstall Compass 2.1.x before running the Compass 2.2.x or higher "Compass.msi". Your configuration files, Compass DB, DICOM image files, and HL7 messages on disk will not be removed. After installing Compass 2.2.x (or higher), the Compass DB and configuration files will be upgraded automatically.

If you attempt to install Compass 2.2.x (or higher) without uninstalling 2.1.x first, the installer will fail. You will have to remove 2.1.x before continuing.

If you currently have a support contract with Laurel Bridge Software, then Compass can automatically upgrade your Compass license over the internet. If you do not have internet access from your Compass server, you will have to manually activate Compass from a different internet enabled computer. See the section on [Installing a License](#) for more details.

IIS Express 7.5 had previously been installed by the Compass installer and will be uninstalled when Compass 2.1.x is uninstalled. For versions of Compass prior to 3.3.0, the IIS Express 8.0 x64 installer is included in your Compass download zip file. Compass 3.3.0 and newer does not use IIS Express. See the [Prerequisites](#) section below.

2.1.4 Upgrading a Failover Cluster to 2.2.x (or higher) from a 2.1.x Version

The steps for upgrading a failover cluster are similar to upgrading a single node; you can, however, keep the services running on one node while the upgrade is occurring on the other node. To upgrade the node not currently running:

- Uninstall Compass 2.1.x
- Install IIS Express 8 (for Compass 3.3.0 and newer, skip this step)
- Install Compass 2.2.x (or higher)
- Start the Compass UI
- The About dialog will be displayed with an "Invalid license". Click the "Activate License..." button
- Fill in the fields and click the "Activate" button
- The About dialog box will now be populated with the updated license info. Click OK.
- After several seconds, the database dialog will appear (this is expected, as the Compass service is running on the other node). Close the database dialog by pressing "Cancel".
- In the Windows Cluster Failover Manager, move the clustered services over to the newly upgraded 2.2.x (or higher) node.
- Start Compass, and verify the UI opens correctly.
- If applicable, open a web browser to verify the Compass web UI is functioning correctly.

- Perform all of the above steps on the other node that is not active.

2.2 Minimum System Specification

Compass runs on dedicated hardware or a virtual machine. Depending on your workflow, your system requirements may differ. For example, a workflow with large/high-volume study counts and/or large image sizes may require improvements to the CPU, RAM, disk, network card(s), etc.

- Intel i5+, 8GB+ RAM, 500GB+ HD 7200+ RPM,
- Windows 10 or newer; Windows Server 2012 R2 or newer.
- SQL Server 2012 x64 or newer.
SQL Express edition may be used for lighter workloads.
The full SQL version must be used for failover cluster configurations.
It is recommended to install the SQL Management Studio as well.

Table of Minimum System Requirements by Compass Version:

Compass:	HL7	Direct	Store	Pro
Windows OS (standard; server) 10; 2012	✓	✓	✓	✓
MS-SQL server + Management Studio	Express	Express	Express	Express
Memory (RAM)	16 GB	16 GB	16 GB	16 GB
Processor	i5	i5	I7	i7
Hard Drive	500 GB	250 GB	500 GB	500 GB
# of network (NIC) cards (Gigabit)	1	1	1	1
Supports high-availability configurations	Yes	Yes	Yes	Yes

* Optional Features/Functions and study volume may dictate server upgrades. MS SQL Standard and/or multiple NICs may be preferred.

Additional system recommendations:

- Enabling various features such as transfer syntax conversion may require upgraded server infrastructure.
- For critical applications, it is recommended that VM resources be reserved.
- If physical access to the server(s) hosting Compass cannot be controlled, full disk encryption technology (such as BitLocker or the use of self-encrypting hard drives) is recommended. See [Appendix A: Section 2.17 Data Storage Confidentiality](#) for more details.

While the definition of an adequate system will depend on the user throughput required, it is recommended that the host system have real-time network bandwidth and supporting processor(s) capable of at least 3x the user defined source traffic loads, plus CPU and RAM capacity to support any peripheral compute requirements like transcoding, filtering, or custom data processing. It must have storage available to contain at least 2x user defined retention loads (studies per day times required days of retention); this is in addition to the storage required for applications and execution, including SQL, O/S, etc.

2.3 Prerequisites

Laurel Bridge Compass utilizes several prerequisite components that must be installed for the application to work.

The following **prerequisites must be installed prior to installing Compass**:

- Microsoft .NET Framework 4.8
- Microsoft SQL Server
- Microsoft SQL Server Management Studio

If Compass is being installed as part of a Windows cluster, the Windows Server 2012 R2(or later) operating system must be installed, and the following prerequisites must be installed prior to installing Compass:

- Microsoft .NET Framework 4.8
- Microsoft SQL Server 2012 x64 (or newer)
- Microsoft SQL Management Studio for SQL Server 2012 x64 (or newer)

2.3.1 Installing SQL Server 2012 Express x64 with Tools

These are instructions for installing SQL Server Express in its most basic configuration for use by Compass. These instructions are valid for Windows 10 and Windows Server 2012 R2. The installation procedure may differ if a different version of SQL Server is installed, if the full version of SQL Server is preferred, or if SQL Server authentication mode must be enabled.

1. Log in to Windows as a user with administrative privileges.
2. Run the [SQL Server 2012Express x64 with tools \(SQLEXPRTW_x64_ENU.exe\)](#) installer.
3. On the [Setup](#) screen select [New installation or add features to an existing installation](#).
4. On the [License Terms](#) screen, accept the license, click the [Next>](#) button.
5. On the [Product Updates](#) screen choose whether to check for updates based on your corporate policy.
6. On the [Feature Selection](#) screen make sure all of the checkboxes are checked for all of the [Instance Features](#). Make sure that the [Management Tools](#) checkboxes are checked, click the [Next>](#) Button.
7. On the [Instance Configuration](#) screen, the defaults should be correct. The named instance should be [SQLExpress](#). Allow it to install in the default location, which should be C:\Program Files\Microsoft SQL Server\
8. On the [Server Configuration](#) screen, the defaults should be fine for the [Service Accounts](#) tab and the [Collation](#) tab defaults.
9. On the [Database Engine Configuration](#) screen on the [Account Provisioning](#) tab, select [Windows Authentication Mode](#).
 - a. The Current user (who must have Administrative Privileges) should be in the list under [Specify SQL Server Administrators](#). If it is not, click the button to [Add Current User](#). Leave the defaults on the other two tabs.
 - b. You must also add the 'NT AUTHORITY\SYSTEM' user. Click the [Add...](#) button and type [System](#) into the text box then click the [Check Names](#) button to add to the list and click OK. You should now see 'NT AUTHORITY\SYSTEM (SYSTEM)' in the list of SQL Administrators. Click the [Next>](#) button.
10. Installation should complete in several minutes.

2.3.2 Installing SQL Server 2014 Express x64 with Tools

These are instructions for installing SQL Server Express in its most basic configuration for use by Compass. These instructions are valid for Windows 10 and Windows Server 2012 R2. The installation procedure may differ if a different version of SQL Server is installed, if the full version of SQL Server is preferred, or if SQL Server authentication mode must be enabled.

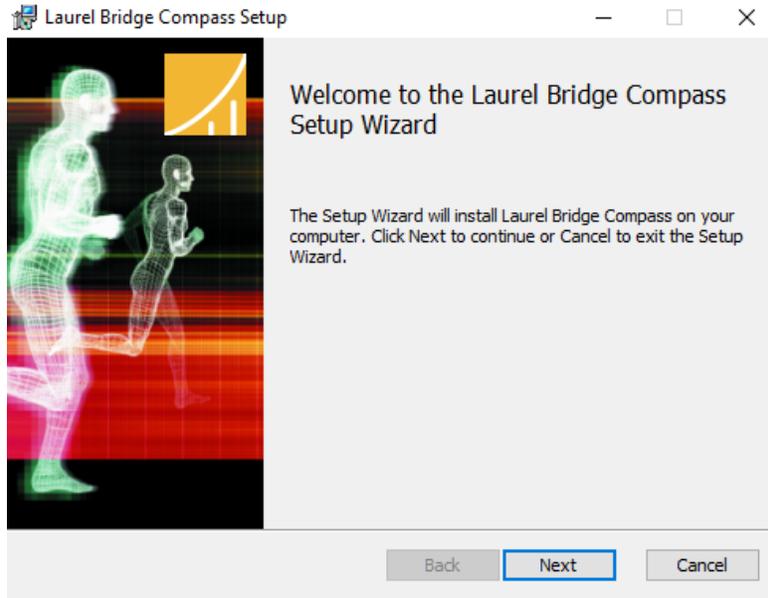
1. Log in to Windows as a user with administrative privileges.
2. Run the **SQL Server 2014 Express x64 with tools (SQLEXPRTW_x64_ENU.exe)** installer.
3. On the **Setup** screen select **New installation or add features to an existing installation**.
4. On the **License Terms** screen, accept the license, click the **Next>** button.
5. On the **Microsoft Update** screen choose whether to check for updates based on your corporate policy.
6. On the **Feature Selection** screen make sure all of the checkboxes are checked for all of the **Instance Features**. Make sure that the **Management Tools** checkboxes are checked, click the **Next>** Button.
7. On the **Instance Configuration** screen, the defaults should be correct. The named instance should be **SQLExpress**. Allow it to install in the default location, which should be C:\Program Files\Microsoft SQL Server\
8. On the **Server Configuration** screen, the defaults should be fine for the **Service Accounts** tab and the **Collation** tab defaults.
9. On the **Database Engine Configuration** screen on the **Account Provisioning** tab, select **Windows Authentication Mode**.
 - a. The Current user (who must have Administrative Privileges) should be in the list under **Specify SQL Server Administrators**. If it is not, click the button to **Add Current User**. Leave the defaults on the other two tabs.
 - b. You must also add the 'NT AUTHORITY\SYSTEM' user. Click the **Add...** button and type **System** into the text box then click the **Check Names** button to add to the list and click OK. You should now see 'NT AUTHORITY\SYSTEM (SYSTEM)' in the list of SQL Administrators. Click the **Next>** button.
10. Installation should complete in several minutes.

2.4 Installation: Compass Application

After installing the prerequisites, the Compass installer (**Compass.msi**) can be run. For machines with an older version installed, this installer will upgrade any previous installation while maintaining any current configuration settings.

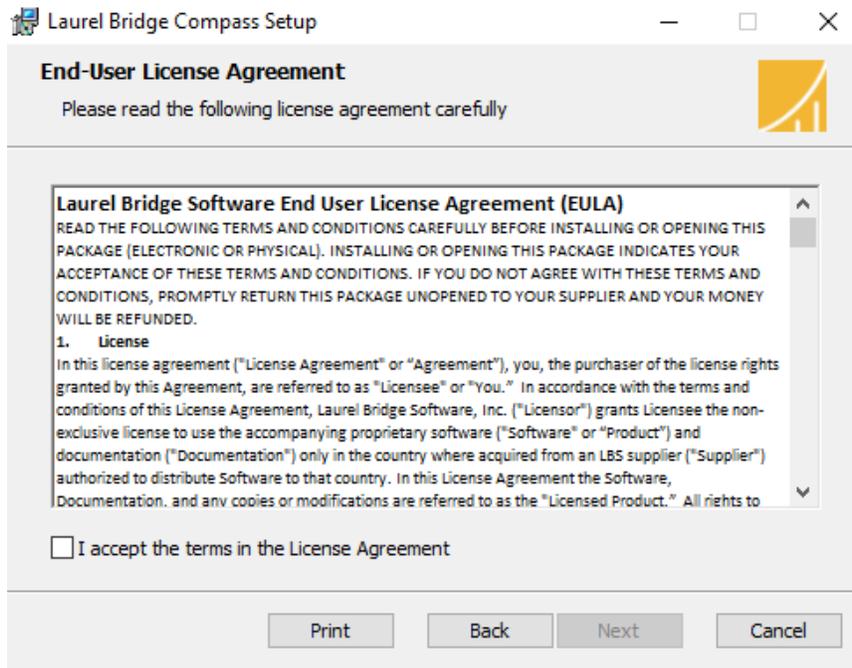
You must have Windows Administrator privileges in order to install Compass correctly.

After launching the Compass installer by double-clicking **Compass.msi**, the user is greeted with the Welcome screen:



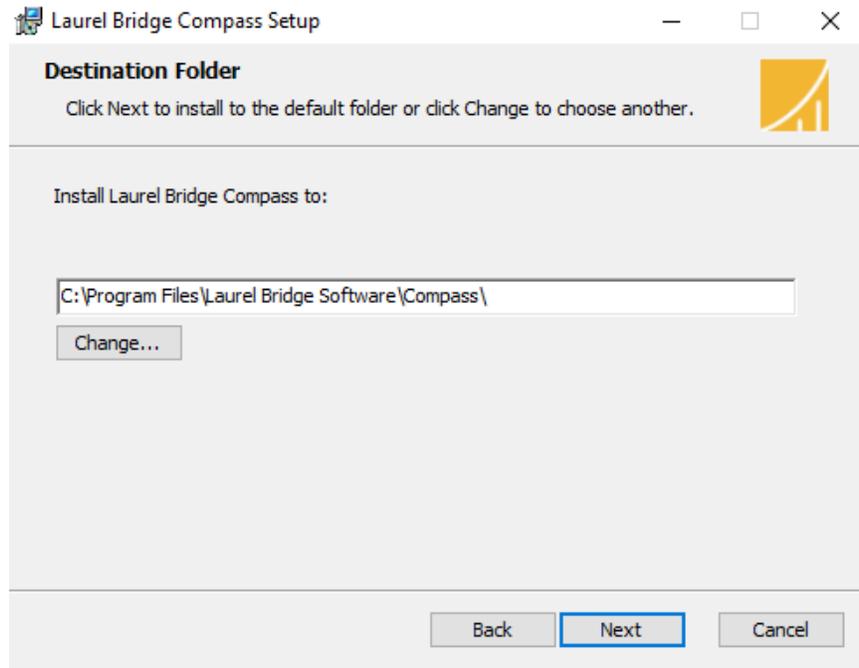
Press the 'Next >' button.

The user is then greeted with the License Agreement screen:



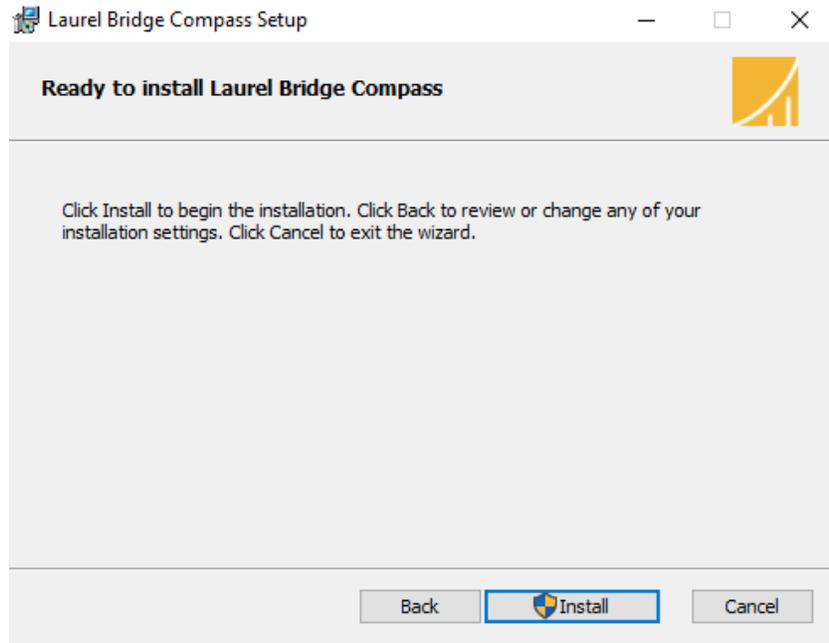
Select the 'I Agree' radio button and then press the 'Next >' button.

The user is then greeted with the Installation Location screen:



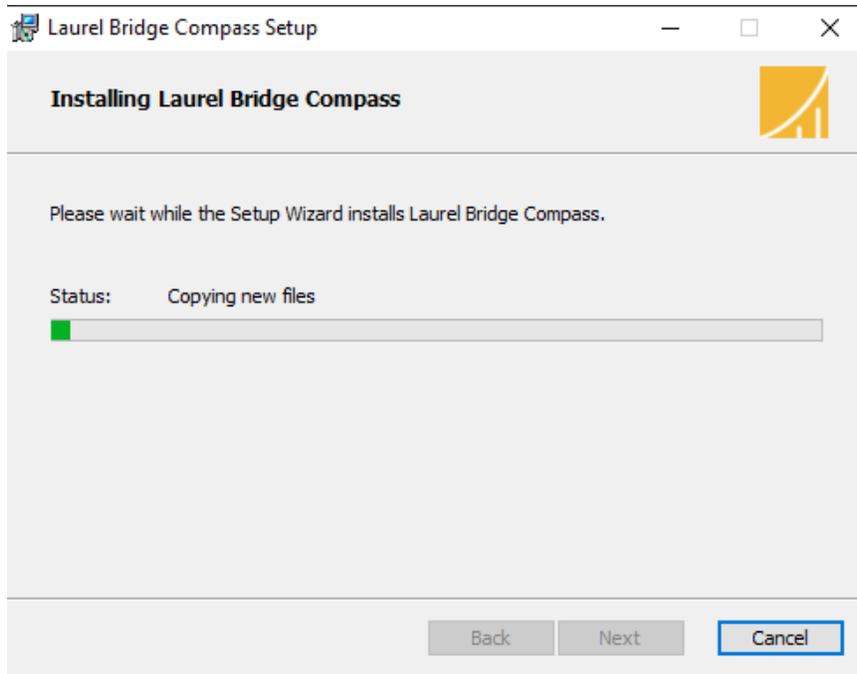
Specify an installation location (the default location is typically the best choice) and then press the 'Next >' button.

The user is presented with a “Ready to install Laurel Bridge Compass” screen:

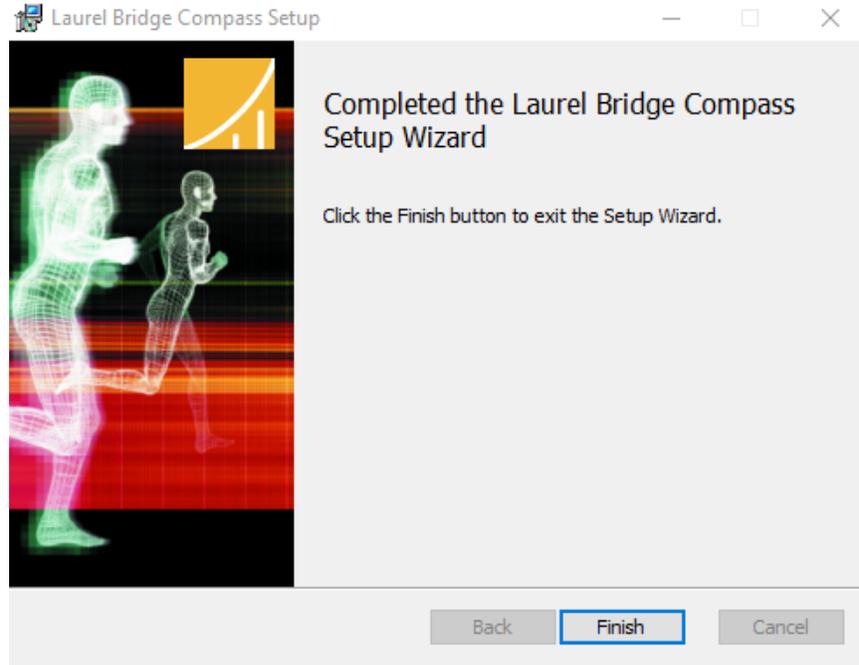


Press the ‘Install’ button to begin installation.

The user is then greeted with an installation progress screen:



After installation completes, the user is then greeted with the Installation Complete screen:



Press 'Finish' to complete the installation.

The computer should be rebooted after installation of Compass.

After rebooting the first step after launching the Compass User Interface is to install a license, which is described in Section 3.2, Installing a License below.

2.5 Configuring Compass Database Connectivity

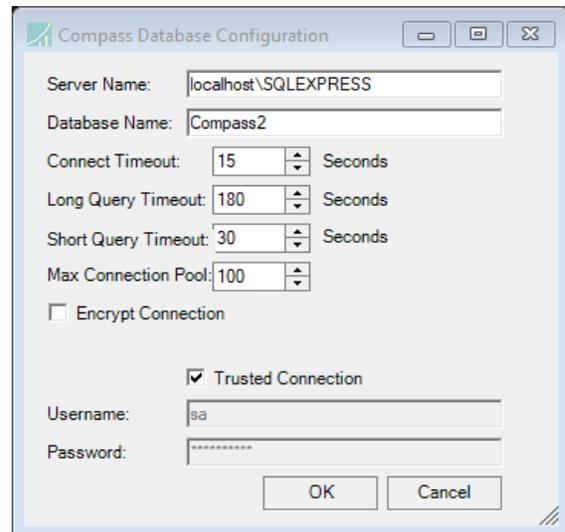
Upon starting Compass for the first time, the Compass Database Configuration dialog will be displayed if Compass is unable to connect to the database using the default values, allowing for entry of the database connectivity information. This dialog will also be displayed whenever Compass encounters errors while attempting to connect to the database.

In order to connect to a SQL Server instance, enter the SQL Server instance name into the **Server Name** field, using the following format:

“<hostname>\<instance_name>”. Enter the SQL

Server database name into the **Database Name** field. The **Connect Timeout**, **Long Query Timeout**,

and **Short Query Timeout** fields can be used to change the default timeout values. The **Encrypt Connection** checkbox can be used to enable encryption on the SQL Server connection (only useful if the SQL Server instance is not on the local machine). The **Trusted Connection** checkbox can be used to toggle whether Windows authentication or **Username / Password** authentication is used to connect to the SQL Server instance. And finally, when using a non-standard TCP port (i.e., other than 1433) to connect to the SQL Server instance, enter a comma followed by the non-standard port number at the end of the **Server Name** field.



If reconfiguration of the database connectivity information is required, the Compass Database Configuration dialog can also be manually started. In order to do this, the Compass service cannot be running. Once the Compass service has been stopped, run the following command (which can be run from an administrator command prompt or by creating a shortcut to the following path+parameter and double-clicking the shortcut):

```
C:\Program Files\Laurel Bridge Software\Compass\CompassClient.exe /dbconfig
```

2.6 Configuring Compass Web Connectivity

When installed, the Compass application is comprised of three processes:

1. **Compass Service**, which runs as a Windows Service; it normally runs with NTAUTHORITY\SYSTEM permissions.
2. **Compass Client**, a Windows Forms-based (UI) application which is used to configure the Compass Service and monitor the Compass jobs; it normally runs with the permissions of the current, logged-in user.
3. **Compass Web**, an optional web interface that allows configured web users to monitor and manage jobs; it normally runs with NTAUTHORITY\SYSTEM permissions.

In a default configuration, configuration changes are made by the user via the Compass Client, which then passes the new configuration information to the Compass Service to make the actual

changes. Thus, no elevated permissions are required for the logged-in user, as the Compass Service makes the configuration changes using its elevated permissions.

3 Getting Started

3.1 Overview

The DICOM and HL7 routing capabilities of Compass are conceptually similar, even though they differ in a few details. In order to route DICOM images/messages, Compass must have in its DICOM configuration at least one Source, one Destination, and one Rule defined. Similarly, in order to route HL7 messages, Compass must have in its HL7 Configuration at least one Source, one Destination, and one Rule defined. The meaning of the terms Source, Destination, and Rule are extremely similar in each case.

For DICOM routing, a Source describes a DICOM SCU that will issue DIMSE messages to Compass. There is no limit on the number of Sources, though at least one must be defined. A Source describes the SCU machine, Application Entity titles, transfer syntaxes, advanced association settings, and data Filters on the images/messages received from the SCU.

A Destination describes an SCP that will receive DIMSE messages from Compass. There is no limit on the number of Destinations, though at least one must be defined. A Destination describes the SCP machine, Application Entity titles, transfer syntaxes, advanced association settings, data Filters on the images sent to the SCP, the sending schedule, and the job activation mode, which describes how DICOM images are aggregated or split in associations. Additionally, there are two special “built-in” destinations. The first is the Hold Queue – images routed to the Hold Queue will not be sent to a remote SCP but will instead be retained in Compass until they are purged or retargeted. The second is the Cache. The Cache is a logical area in Compass where images are organized by Patient/Study/Series/Image and can be queried via standard DICOM Query/Retrieve mechanisms. The Cache is discussed more in a later section.

A Rule describes the criteria a DICOM message must meet in order to be sent to the specified Destination(s). There is no limit on the number of Rules, though at least one must be defined. A Rule contains one or more conditions: conditions based upon the association information, conditions based upon DICOM tag information, or custom user-defined conditions. Each individual Rule can be specified such that any or all of the conditions must be met in order for the DICOM message to be tasked for the specified Destination(s). Each individual rule also indicates whether it is a **Store and Forward** rule or **Direct** rule, which controls the behavior when routing a message which matches that particular rule.

For HL7 routing, a Source describes an HL7 Sender that will send HL7 messages to Compass. There is no limit on the number of Sources, though at least one must be defined. A Source describes the sending machine, Sending Application, Sending Facility, Compass listening port, and advanced connection settings which apply for messages from that Source.

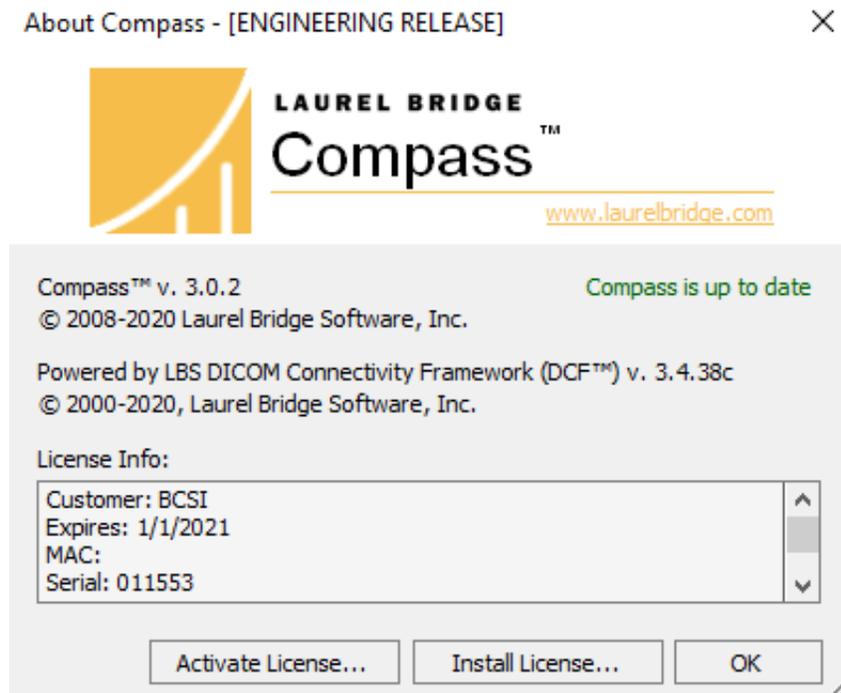
A Destination describes an HL7 Receiver that will receive HL7 messages from Compass. There is no limit on the number of Destinations, though at least one must be defined. A Destination

describes the machine, ordering logic, sending schedule, and other advanced settings for the particular HL7 receiver.

A Rule describes the criteria an HL7 message must meet in order to be sent to the specified Destination(s). There is no limit on the number of Rules, though at least one must be defined. A Rule contains one or more conditions: conditions based upon the network connection information, conditions based upon HL7 message content, or custom user-defined conditions. Each individual Rule can be specified such that any or all of the conditions must be met in order for the HL7 message to be tasked for the specified Destination(s).

3.2 Installing a License

The first step after launching Compass is to install a license. When the **About Laurel Bridge Compass** dialog appears, there are two options to install a license. Information on the status of the license, including when it expires, can be found on this dialog available at any time by clicking **Help > About Laurel Bridge Compass**.



3.2.1 Installing a license file

Clicking the **Install License...** button will open a file searcher dialog allowing you to browse to a fully activated license file. You will use this button to install a license if a fully activated license was provided to you via email (for example, a MAC-based license) or you downloaded from www.laurelbridge.com or you performed a web activation.

Note: On Windows Server you must copy the license file to the local computer before installation. You cannot install the license from a network share or mapped network drive.

3.2.2 Activating a license over the internet

Clicking the **Activate License...** button will open the **License Activation** form. You will need to provide a **16-digit Product Serial Number** to activate your license (if you are upgrading from a Compass 2.1.x version and please see below for instructions on how to obtain your **Product Serial Number**.) All fields on the form must be filled out. If the Compass server does not have internet access you will need the **16-digit Activation Request Code (ARC)** from the **License Activation Form**. You can proceed to https://www.laurelbridge.com/product_activation.php and enter your **16-digit Product Serial Number** and the **16-digit Activation Request Code (ARC)**. After completing the steps on the web activation form you can download a fully activated Compass license. You can then use the **Install License...** button on the **About Compass** dialog to install your activated license.

The screenshot shows a web browser window titled "License Activation" with a close button (X) in the top right corner. The form is organized into three sections, each with a question mark icon to the left of the labels:

- Product Section:**
 - Name: Ver:
 - Product S/N:
 - ARC:
 - MAC:
- User Section:**
 - Site:
 - Contact:
 - Email:
 - Host Name:
- Maintenance Section:**
 - Contact:
 - Email:
 - Phone:

At the bottom of the form are two buttons: "Lookup" and "Activate".

If you are upgrading from Compass 2.1.x version and you previously had a MAC based license Compass will attempt to connect to the Laurel Bridge license server and find your 16-digit **Product Serial Number**. If it cannot reach the server you will be presented with an error message describing which fields need to be entered on the https://www.laurelbridge.com/product_activation.php to activate your

license. If you only have a MAC address you will need a valid login to www.laurelbridge.com to proceed. If you need assistance, please email support@laurelbridge.com.

3.2.3 Activating a license that added the HL7 Edition

The Compass standard edition license provides DICOM routing capabilities. You can upgrade your Compass license to provide capabilities for HL7 messages in addition to DICOM messages. After following the procedure in sections 3.2.1: [Installing a license file](#) or 3.2.2: [Activating a license over the internet](#), you must stop the Compass Client and stop and restart the Compass Service to initialize and activate the HL7 Server within Compass.

4 Configuration for Routing DICOM

This section of the user manual will focus on how to configure Compass for routing of DICOM images/messages, including configuration of Sources, Destinations, Rules, Job Actions, Listeners, and Filters. Note that many of these configuration areas have a Description field; this field is a free-form text box which can be used to hold user notes about the configuration item.

Description:

4.1 Creating a Source

A new Source can be created by selecting **Edit > DICOM Options...** from the menu and then selecting the **Sources** pane. Press the green plus button located under the Sources list. This will pop up a context menu with three choices: **DICOM Source**, **Hot Folder Source**, and **DICOMweb Source**. The list of Sources may be reordered by selecting a Source and then pressing the up or down arrows next to the green plus and red minus buttons.

4.1.1 DICOM Source

A **DICOM Source** is any device that will send DICOM images/messages to Compass via an association. Selecting **DICOM Source** will create a new Source with the name “New Dicom Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name. When an SCU requests an association with Compass, the Sources list is processed from top to bottom and the first match of type **DICOM Source** found is used; therefore, the ordering of the Sources may be important if potentially more than one could match.



Once a **DICOM Source** has been created the next step is to configure its settings.

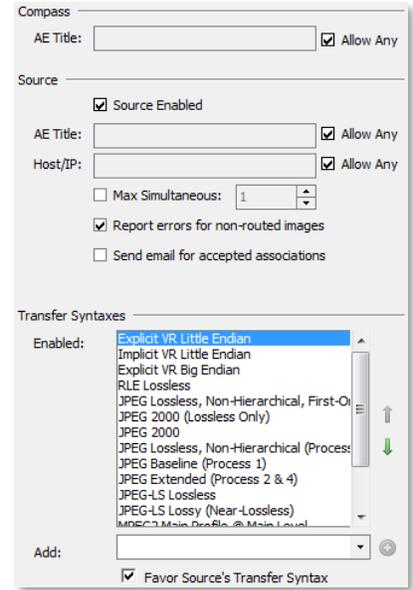
4.1.1.1 Settings

Under the **Compass** heading, the AE Title that Compass will accept is listed. The **Allow Any** checkbox can be selected to allow any value for that field. For installations with a Compass Basic license, the **AE Title** must be specified.

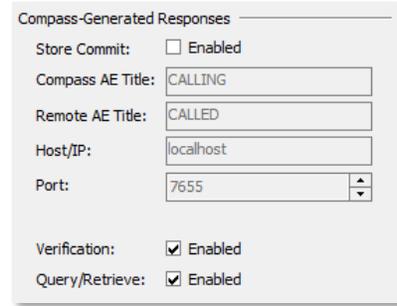
Under the **Source** heading, settings include configuration options for the remote **AE Title** (AE title of the client), the remote **Host/IP**, the Enabled status of the Source, an option to limit the number of concurrent associations, and the type of responses that are sent back to the Source for non-routed images. The **Allow Any** checkbox following the **AE Title** and **Host/IP** can be selected to allow any value for that field. For installations with a Compass Basic license, the **AE Title** must be specified. Select the **Source Enabled** checkbox to enable the specified Source; deselect it to temporarily refuse any connections from the specified Source. To limit the number of concurrent associations allowed by this Source check the **Max Simultaneous** checkbox and specify the desired number.

The type of response from Compass that will be sent to the Source for non-routed images (submitted via C-STORE request) can also be configured. Images fail to get routed to a Destination if they don't match any of the defined Rules. Compass's default behavior is to report an error back to the Source for a non-routed image; this is indicated by selecting the checkbox **Report errors for non-routed images**. By deselecting this checkbox, Compass will always report back to the Source that it received the image successfully. This behavior is desirable in a situation when Compass should not forward all of the images sent to it by this particular Source. For example, if all Ultrasound images with a particular Referring Physician need to be sent from a PACS to another Destination, but the PACS can only send all of its Ultrasound images and not simply a subset of them, then, by leaving the Report errors for non-routed images checkbox unchecked and defining a Rule to match the Referring Physician, only the desired images will be sent to the Destination.

Under the **Transfer Syntaxes** heading, the list of transfer syntaxes that Compass will accept for the specified DICOM Source can be configured. Add the desired **Transfer Syntaxes** to the list of accepted Transfer Syntaxes by selecting the desired Transfer Syntax in the combo box and adding it by pressing the provided button. After building up the list with the desired Transfer Syntaxes, place them in the desired order by selecting each one and moving it up and down in the list by pressing the up and down arrows. The supported Transfer Syntax list is evaluated top-to-bottom when a Source presents its requested presentation context list. If the Favor Source's Transfer Syntax checkbox is checked, then the Transfer Syntax list ordering proposed by the Source will be given priority; otherwise, the list as specified in Compass will be given priority.

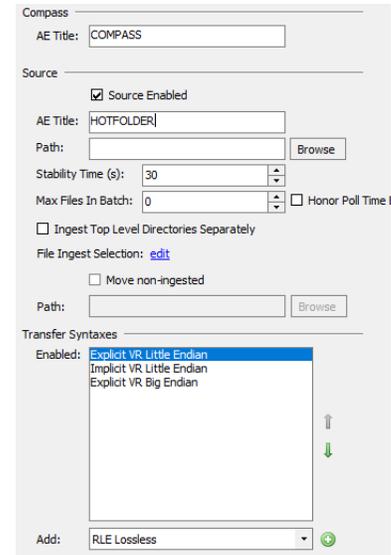


Under the **Compass-Generated Responses**, a Source can be configured to handle a few specific DICOM messages. If the **Storage Commit Enabled** checkbox is checked, Compass will respond to all store commit requests from the given Compass Source. Compass will respond with an “accept” response to the specified **Host/IP** and **Port** using the specified **Compass AE Title** and **Remote AE Title**. Similarly, if the **Verification Enabled** checkbox is checked, Compass will respond to verification messages from the Source. If the **Query/Retrieve** checkbox is checked, Compass will respond to C-FIND requests with responses based on the contents of the Compass Cache. In all these cases, if the **Enabled** checkbox is not checked, any received Store Commit/Verification/Query messages will be processed via the Rules which are configured and may thus be **Direct**-routed to a destination.



4.1.2 Hot Folder Source

A “Hot Folder Source” is a Windows filesystem folder that Compass will continually monitor for DICOM files (and, optionally, non-DICOM files), ingesting and filtering them after reaching a configurable stability time. Selecting **Hot Folder Source** will create a new Source with the name “New Hotfolder Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name.



If non-DICOM files are ingested, Compass will create instances of a private SOP class for each such file, adding specific DICOM header tags retrieved from the first DICOM file found in any batch. Support for non-DICOM files requires a license. Contact Laurel Bridge to enable this feature.

Note: it is the user’s responsibility to ensure that any and all data that may be transferred or transmitted using Compass is free of viruses, worms, Trojans, bots, or other code of a destructive or undesirable nature, collectively known as malware. Compass has no responsibility to scan or inspect in any way the data transferred or transmitted for the presence of malware and Laurel Bridge Software, Inc. has no liability of any kind associated with the transfer or transmission of such data.

Compass will delete the instances from the hot folder after they have been copied to Compass’ local instance repository.

Please note that due to the nature of **Immediate Mode Destinations** and their requirement that all possible inbound presentation contexts be known at the start of ingest, images ingested via a hot folder cannot be routed to an **Immediate Mode Destination**. Any **Immediate Mode Destinations** that were matched via the rules will be skipped over, and an error message will be written to Compass’s application log file.

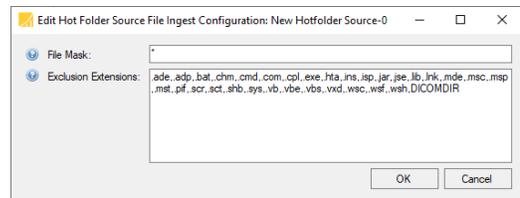
Please also note that images ingested via a hot folder can only match **StoreAndForward** rules; rules marked as **Direct** will be skipped. If the Compass license does not have the **StoreAndForward** attribute enabled and a hot folder source is defined, the configuration is incompatible with the license and will not be saved. Instead, an error message will be presented to the user indicating the incompatibility.

Once a **Hot Folder Source** has been created the next step is to configure its settings.

4.1.2.1 Settings

Under the **Compass** heading, the **AE Title** that Compass will apply to images ingested from the specified hot folder is listed. Even though the instances aren't coming from an association, Compass provides the ability to specify the AE titles on the hot folder for later use, such as making a routing decision based on one or both of the AE titles.

Under the **Source** heading, settings include configuration options for the remote **AE Title** (the calling AE title), the **Path** which should be monitored for DICOM instances, and a **Stability Time** which indicates how long to wait for the folder contents to become stable. There are batch control settings: **Max**



Files in Batch, **Honor Poll Time Between Batches**, and **Ingest Top Level Directories Separately** (see **Appendix F: Hot Folder Basics / Definitions**). Additionally, **File Ingestion Selection** restricts the hot folder to ingest or exclude specifically named files. The **File Ingestion Selection** link presents a dialog with two separate controls. The first, **File Mask**, is a regular expression that allows the user to select which files should be considered for ingestion. The second, **Exclusion Extensions**, provides a list of filename endings which will specifically be excluded. Exclusion takes precedence over inclusion. So, for example, if the user picks “*” for the **File Mask** (meaning “include everything” but uses the default exclusion list, no files with the default extensions (“.exe”, “.bat”, etc.) will be ingested.

There is also a checkbox that, if checked, moves non-ingested files out of the Hot Folder and into another user specified folder for later inspection.

Under the **Transfer Syntaxes** heading, the list of transfer syntaxes that Compass will accept for the specified Hot Folder Source can be configured. Add the desired **Transfer Syntaxes** to the list of accepted Transfer Syntaxes by selecting the desired Transfer Syntax in the combo box and adding it by pressing the provided button. Compass will only ingest DICOM images that have a transfer syntax listed in the **Enabled** list.

Under the **Advanced** heading, **Filters** can optionally be configured (see below) for the Source. There is also a **Logging** feature which, if set to **On**, logs a descriptive report describing instances that failed to be ingested as well as instances that were successfully ingested. The **Settings** dialog specifies the configuration of the Source's **Activity Threshold**, which affects the Status column on the **Source/Destination Status** dialog. When sending an outbound job, the IP address Compass will use can be specified by the **Source** via the **Outbound IP** combo box, if and only if the Destination that the job is targeted for has its **Host/IP** combo box set to Default. There is also a checkbox labeled **Send email on ingest failure**. If this is checked, an email will be sent to the

notification recipients listed on the **Notifications** tab if a file fails to be ingested successfully. The checkbox for **Ingest non-DICOM files** indicates whether this source supports ingestion of non-DICOM files in addition to DICOM files. If this box is unchecked, the Hot Folder Source will scan the files in its Path folder and only ingest files which it recognizes as DICOM files, leaving the other files untouched (they may, of course, then be moved if the **Move non-ingested files** checkbox is checked). When non-DICOM file ingestion is enabled and a batch of files is ingested via a Hot Folder Source, the **Ingestor** specifies the path to the code file which implements the ingestor interface. If this value is blank (which it is by default), Compass will use the default ingestion scheme. See the Custom Code examples later in this manual for examples of an alternate **Ingestor**.

4.1.2.2 Zip File Ingestion

When scanning the contents of a hotfolder, if a valid zip file with an extension of “.zip” is encountered, Compass will handle it in one of two ways. If the **Ingest non-DICOM files** checkbox is checked, Compass will treat the file as a non-DICOM file and move it as-is. If unchecked, Compass will extract the zip file and attempt to ingest the contents of the zip file as its own ingestion batch.

Please note that the decision to ingest the contents of a zip file is a function of the stability time of the zip file itself; the timestamps of the files contained in the zip file do not factor in to the stability time ingestion decision.

4.1.3 DICOMweb Source

A **DICOMweb Source** is any device that will send DICOM images/messages to Compass via a DICOMweb HTTP connection. Specifically, a **DICOMweb Source** will accept requests using the Store Transaction of the Studies Service and Resources protocol (commonly referred to as STOW-RS) of the DICOM PS 3.18 – Web Services specification (commonly referred to as DICOMweb). STOW-RS requests can match both **Store and Forward** or **Direct** rules.

If **Direct** mode is enabled, a **DICOMweb Source** can also accept and direct-route requests using the Query Transaction of the Studies Service and Resources protocol (commonly referred to as QIDO-RS) and the Retrieve Transaction of the Studies Service and Resources protocol (commonly referred to as WADO-RS). QIDO-RS and WADO-RS requests can only match **Direct** rules.

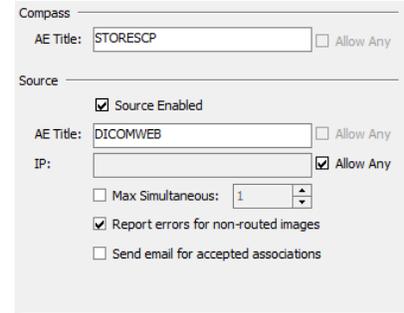
Selecting **DICOMweb Source** will create a new Source with the name “New DicomWeb Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name. When a DICOMweb HTTP connection is received, the Sources list is processed from top to bottom and the first match of type **DICOMweb Source** found is used; therefore, the ordering of the Sources may be important if potentially more than one could match.

Once a **DICOMweb Source** has been created the next step is to configure its settings.

4.1.3.1 Settings

Since the settings for sources of type **DICOMweb Source** are very similar to those of sources of type **DICOM Source**, this section will only describe the differences between the two. Please see the section above for configuring sources of type **DICOM Source** for details about shared configuration parameters.

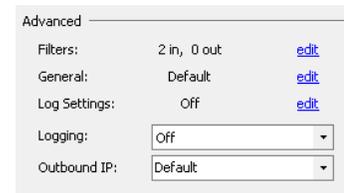
One big difference between a **DICOM Source** and a **DICOMweb Source** is that the DICOMweb protocol does not have any concept of AE titles. Thus, both the Compass **AE Title** and Source **AE Title** cannot be used to match incoming connections. The values entered here are only for connection naming purposes, as they will be used in the pseudo-association record that is created during a STOW-RS import session. This also explains why the **Allow Any** checkboxes are disabled for DICOMweb Sources.



Since the Store Commit SOP class does not apply to DICOMweb connections, all the controls related to **Compass-Generated Responses** are disabled, except for **Query/Retrieve**. If the **Query/Retrieve** checkbox is checked, Compass will respond to QIDO-RS and WADO-RS requests with responses based on the contents of the Compass Cache.

4.1.4 Advanced Settings

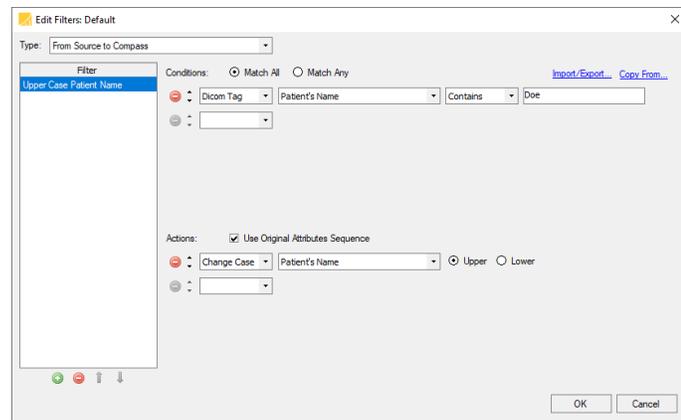
Under the **Advanced** heading, a user can give more in-depth control over how Compass communicates with a Source. Log Settings are also configured for the Source.



4.1.4.1 Filters for Sources

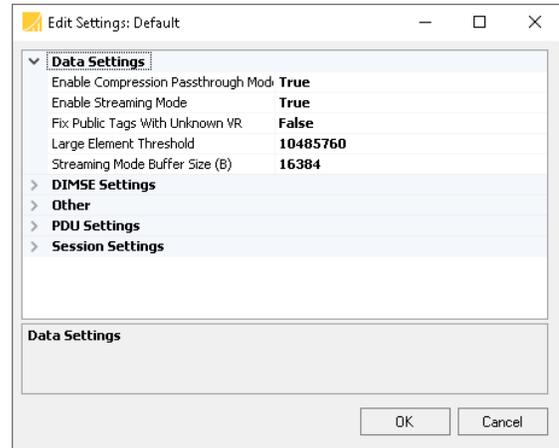
Filters can be created for any Source.

The list of Filter names may be reordered by selecting a Filter and then pressing the up or down arrows. Filters in the list will be applied in order from top to bottom, so ordering is important. **See Section 4.7 Filters** for a more in-depth description of filtering capabilities.



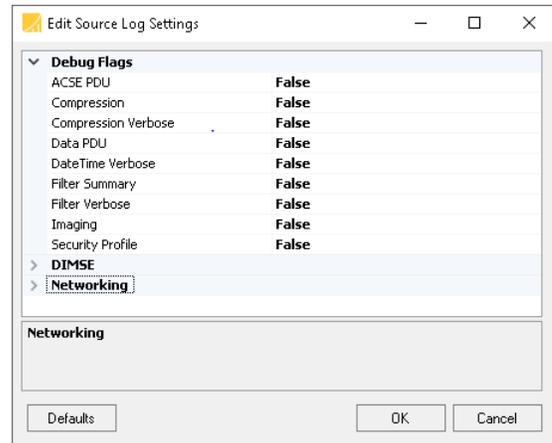
4.1.4.2 General Settings

The **edit** link next to **General** selection allows the user to specify parameters for the Data, DIMSE, Other, PDU and Session Settings. The Data Settings, DIMSE Settings, Other and PDU Settings affect how incoming DICOM messages and outgoing responses are processed by Compass to and from the Source. The Session Settings provides a Session Name to identify a particular session in the Compass Logs.



4.1.4.3 Log Settings

Verbose DICOM logging can be turned on or off on a per Source basis by selecting On or Off for Logging. When Logging is turned on for the Source, the Log Settings define which debug flags, DIMSE message activity, and networking activity are enabled for verbose logging. Only the properties with a True value will generate verbose logs for the Source.



4.1.4.4 Logging

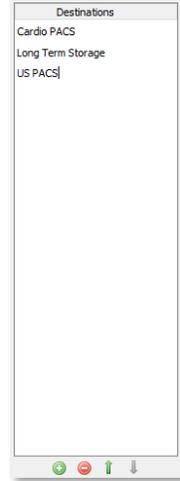
Verbose DICOM logging can be turned on or off on a per Source basis by selecting On or Off in the **DICOM Logging** chooser.

4.1.4.5 Outbound IP Configuration

When sending an outbound job, the IP address Compass will use can be specified by the **Source** via the **Outbound IP** combo box, if and only if the Destination that the job is targeted for has its **Host/IP** combo box set to Default.

4.2 Creating a Destination

A new Destination can be created by selecting **Edit > DICOM Options...** from the menu and then selecting the **Destinations** pane. Press the green plus button located under the Destinations list. This will create a new Destination with the name “New Destination-?” where “?” is the next available number, starting at 0. Destination names are customizable and can be modified at any time by clicking on the name. The list of Destinations may be reordered by selecting a Destination and then pressing the up or down arrows directly below the Destinations list. Unlike Sources, there is no priority associated with Destination ordering; it is merely provided as a convenience.



Once a Destination has been created the next step is to configure its settings.

4.2.1 Compass Settings

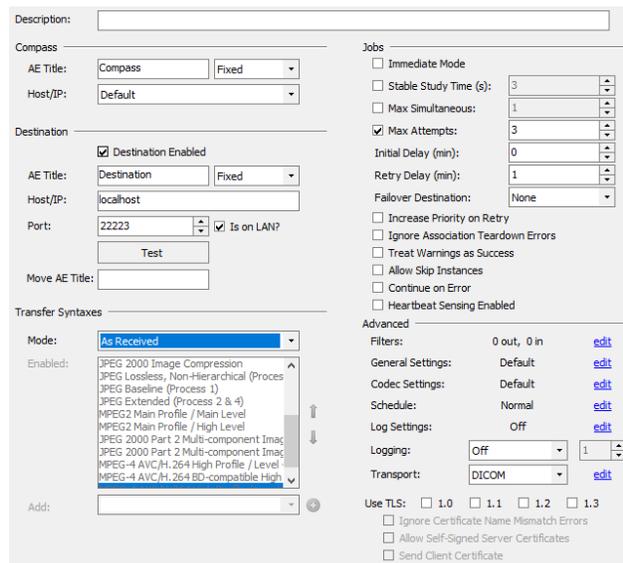
Under the **Compass** heading, settings include configuration options for the **AE Title** (Application Entity) and **Host/IP**.

4.2.2 Destination Settings

Under the **Destination** heading, settings include options for the **AE Title**, the **Host/IP**, the **Port** number, a **Test** button for testing the connection, an **Is on LAN?** checkbox, and the enabled status of the Destination. Selecting the **Destination Enabled** checkbox will allow jobs to be sent to the specified Destination; deselect it to temporarily stop sending jobs to the specified Destination. Checking or unchecking the **Is on LAN?** checkbox will optimize network buffer sizes for that particular network topology.

For both the **Compass** and **Destination AE Title**, values can be set with a fixed value by selecting **Fixed** in their corresponding combo boxes. Alternately, the AE titles can be set to either **Source Calling** or **Source Called**; this will use the calling or called AE title from the inbound association that created the job for this Destination. In the rare case that an outbound job is made up of images that came from different clients using different AE titles, the AE titles from the first image’s association will be used.

Once values have been specified for the Local AE Title, the Remote AE Title, the Host/IP, and the Port, then the **Test** button can be clicked to issue a DICOM Verification request to the specified Destination. A green check next to the Test button indicates a successful connection; a



red X indicates a connection could not be made, or the connection was refused. Hover the mouse pointer over the X to read a tooltip with a message that further describes the failure.

4.2.3 Transfer Syntax Settings

Under the **Transfer Syntaxes** heading, the user can configure the list of allowable transfer syntaxes that Compass will negotiate with the specified Destination. Add the desired transfer syntaxes to the list of accepted Transfer Syntaxes by selecting the desired transfer syntax in the combo box and adding it by pressing the provided button. After building up the list with the desired Transfer Syntaxes, place them in the desired order by selecting each one and moving it up and down in the list by pressing the up and down arrows. Selecting one of the options in the **Mode** combo box will also affect how Transfer Syntaxes are requested:

- **As Received or Destination Preferred** – attempts to send instances to the Destination in the transfer syntax they were received; otherwise attempts to send via the Destination’s preferred transfer syntax from the list of enabled transfer syntaxes.
- **As Received or Compass Preferred** – attempts to send instances to the Destination in the transfer syntax they were received; otherwise attempts to send via Compass’ preferred transfer syntax from the list of enabled transfer syntaxes.
- **As Received** – attempts to send instances to the Destination in the transfer syntax they were received and will fail otherwise.
- **Destination Preferred** – attempts to send instances to the Destination via the Destination’s preferred transfer syntax from the list of enabled transfer syntaxes.
- **Compass Preferred** – attempts to send instances to the Destination via Compass’ preferred transfer syntax from the list of enabled transfer syntaxes.

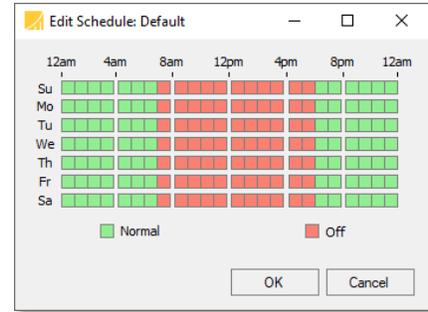
4.2.4 Jobs Settings

Under the **Jobs** heading, settings include options which affect how and when jobs are sent to the Destination.

The **Immediate Mode** checkbox affects several of the other Jobs settings. When **Immediate Mode** is checked, **Stable Study Time (s)** is unchecked and disabled, and **Allow Skip Instances** and **Continue on Error** are checked and disabled.

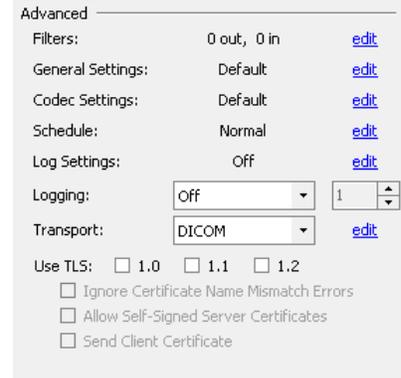
When the **Immediate Mode** checkbox is checked, images routed to this Destination will be sent prior to the closing of the inbound association on which the images originated. Jobs destined for an **Immediate Mode** Destination will show up in Compass’ main jobs table with a State of **Pending/Running**, to reflect both the **Pending** (i.e. currently receiving) and **Running** (i.e. currently sending) status of the job. If an outbound error occurs while processing a **Pending/Running** job (e.g. the Destination unexpectedly terminates the association), the job’s state will convert to **Pending**, and will from thereon be treated as a non-**Immediate Mode** job. N.B. **Immediate Mode** only applies to jobs created by messages which match rules which are **Store and Forward**; this setting is different from, and has no effect on, messages which match rules which are **Direct** (because, by definition, messages which match **Direct** rules do not create any jobs).

If the **Stable Study Time** checkbox is selected, images from the same study that arrive on separate associations will be combined and sent out on a single association, once the user-specified stable time (specified in seconds) has been reached. If unchecked, outgoing jobs will only contain images from the same study that were received in the same association. The **Max Simultaneous** checkbox allows the number of concurrent associations to the specified Destination to be limited to the specified value. The **Initial Delay** setting will delay jobs to this Destination by the specified number of minutes. The **Max Attempts** checkbox defines the maximum number of send attempts. If unchecked, the job will retry indefinitely. The **Retry Delay** will cause an unsuccessful job to be retried after the specified number of minutes. The **Failover Destination** setting allows, for **Store and Forward**, a copy of the original job to be sent to the specified failover destination (if specified) once the original job has reached its **Max Attempts** without succeeding. Note that it is possible to configure an infinite loop of failover jobs, as Compass does not attempt to restrict the failover configuration. Additionally, in **Direct** mode, the **Failover Destination** provides an alternate destination to which Compass should connect if this Destination is network-unreachable (or if its heartbeat is down and **Heartbeat Sensing Enabled** is checked). Checking the **Ignore Association Teardown Errors** checkbox will result in the job being successfully sent if Compass receives all successful DIMSE responses from the Destination, even if the association is not properly released. This functionality is occasionally necessary because some store servers do not properly respond to an association release request. Checking the **Treat Warnings as Success** checkbox allows Compass to treat a job as successfully sent even if one of the DIMSE response statuses specified a warning instead of success. Checking the **Allow Skip Instances** checkbox will cause Compass to skip over images for which there was no accepted presentation context id and still send the remaining images. Checking the **Continue On Error** checkbox will cause Compass to continue sending the remaining images in a job even if an error status is received for an image. If either the **Allow Skip Instances** or **Continue on Error** setting is checked, a job that completes with at least one successfully sent image will have its job state set to **Sent**, with a description of the image errors in the jobs table **Message** column. More detailed information is available in the **Job Report**. Checking the **Heartbeat Sensing Enabled** checkbox will cause Compass to verify that the Destination is available prior to attempting to send jobs to it. If the Destination is unavailable, Compass will queue up the jobs for the Destination without attempting to send them. If and when the Destination becomes available, Compass will begin sending the queued jobs. Compass issues a DICOM Verification request and inspects the reply (or lack thereof) to determine if the Destination is available.



4.2.5 Advanced Settings

Under the **Advanced** heading, a user can give more in-depth control over how Compass communicates with a Destination. Log Settings are also enabled and configured for communication with the particular Destination.



4.2.5.1 Filters for Destinations

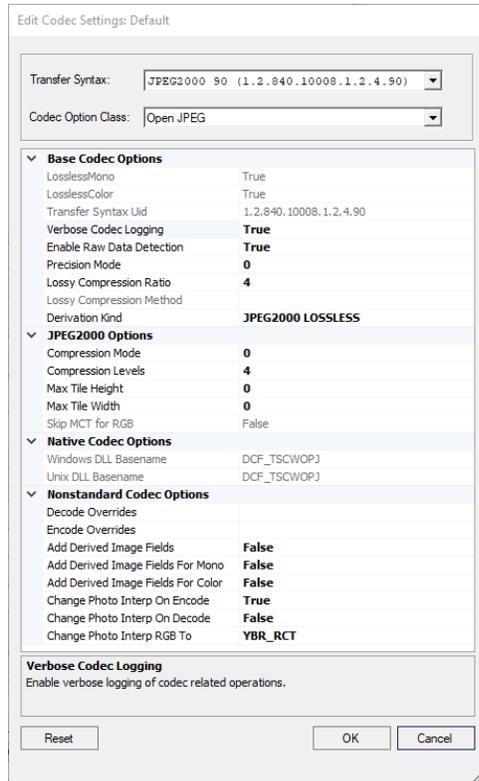
Filters are configured in the same fashion as Sources.

4.2.5.2 General Settings

General Settings are configured in the same fashion as Sources.

4.2.5.3 Codec Settings

The Codec Settings dialog allows for the advanced configuration of the image codecs used to encode instances sent to the current destination.



First, select the Transfer Syntax of the instances to be encoded, such as JPEG2000 90 (1.2.840.10008.1.2.4.90).

Second, determine the codec to use for encoding all .90 instances sent to this destination. In the picture shown, Open JPEG is selected.

Finally, make the necessary reconfiguration based on the property grid presented. Once finished, simply press ‘OK’ to return to the destination configuration form. ‘Cancel’ will discard any changes made and return to the destination configuration form unchanged.

‘Reset’, when enabled, indicates there are changes present for the current Transfer Syntax that differ from the default settings. Pressing ‘Reset’ will discard any changes for the selected Transfer Syntax, returning to the default options for the selected Transfer Syntax only.

Note: Care should be taken when adjusting these codec options for any of the presented Transfer Syntaxes. Irreversible damage may be done to any instance encoded

with improperly configured codec options. Take special care with any option in the **Nonstandard Codec Options** group.

4.2.5.4 Schedule

The **Schedule** specifies the time at which jobs are allowed to be sent to the Destination. A green box for the given day and time means jobs may be sent at that time; a red box means that jobs may not be sent at that time. This affects only the starting time of the job as a whole; a currently running job will not stop sending images once it has begun to transmit them. Each hour block represented by a green or red square may be toggled individually; alternatively, right-clicking on

the schedule allows the schedule to be set to a commonly used setting. Be aware that choosing one of these predefined settings will replace the currently specified schedule.

4.2.5.5 Log Settings

Log Settings are configured in the same fashion as Sources.

4.2.5.6 Logging

Verbose DICOM logging can be turned on or off on a per Destination basis by selecting On, Off, or After Failures in the **DICOM Logging** chooser. Selecting After Failures requires the number of failures to occur before logging is enabled.

4.2.5.7 Transport

The **Transport** chooser determines how jobs will be sent to this destination. There are several possible values for **Transport**:

- The default value for Transport is, quite naturally, **DICOM**. The “edit” link next to this selection allows the user to specify parameters specifically for **Direct**-routed messages for this destination (these settings do not apply to matched **Store and Forward** rules.)

For **QIDO-RS**:

The **Resource Path** configures the RESTful service study-level resource path. This is typically the value “/studies”. The **Response Media Type** configures the expected media type (i.e., the “Accept” header media type) for the HTTP response. Compass supports the “application/dicom+json”, “application/dicom+xml”, “application/json”, “application/xml”, and “multipart/related; type=application/dicom+xml” response media types.

For **WADO-RS**:

The **Resource Path** configures the RESTful service study-level resource path. This is typically the value “/studies”. The **Preferred Transfer Syntax** configures the preferred transfer syntax encoding for the response data set(s). And finally, the **Response Media Type** configures the expected media type (i.e., the “Accept” header media type) for the HTTP response. Compass supports the “application/dicom+json”, “application/dicom+xml”, “application/json”, “application/xml”, and “multipart/related; type=application/dicom+xml” response media types.

For **Authentication**:

If enabled, the **Platform** configures the type of authentication to be performed. Currently, Google Cloud Platform (GCP) is the only supported platform. The **Authentication Method** configures the type of GCP authentication performed. This can be either “Json Key File” or “Certificate”. For “Json Key File”, the **File Path** and **Scope** can be configured. For “Certificate”, the **File Path**, **Password**, **Auth. URL**, **Email**, and **Scope** can be configured. For both types of authentication, the scope is usually the following URL:

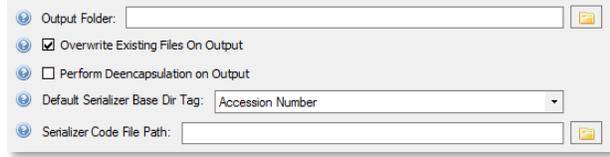
<https://www.googleapis.com/auth/cloud-healthcare>

Also note that the Resource Path for GCP usually starts with:

`/v1/projects/{project}/locations/{location}/datasets/{dataset}/dicomStores/{dicomStore}/dicomWeb`

- DIMSE Relay for more information.

- The **TOPS**, or **Throughput Optimization Protocol** value, means that Compass will use proprietary protocols to transfer jobs to this Destination. Compass **TOPS** requires that the Destination be another instance of Compass version 2.1.0 or greater. **TOPS** provide throughput enhancements for jobs with many instances as well as enhanced job restart capability in the event of an error mid-transfer (such as a temporary network failure).
- The **DISK FOLDER** value means that Compass will write the instances to a specified Disk Folder location.



When **DISK FOLDER** is selected, a configuration dialog will appear which allows the user to select the **DISK FOLDER** options. The **Output Folder** indicates where on disk to write the instances in the job. Overwrite indicates whether to replace existing files on write. **Perform Deencapsulation** tells Compass whether to extract the original content from files that were originally ingested by another Compass as non-DICOM files. **Default Serializer Base Dir Tag** tells Compass what subdirectory should receive any files which are being written out as DICOM. For example, if the **Output Folder** is “C:\MyDir”, **Accession Number** is the **Default Serializer Base Dir Tag**, and the job has an **Accession Number** of “123”, all DICOM instances will be written to a directory structure within “C:\MyDir\123”. Finally, **Serializer Code File Path** points to a custom implementation for how files should be written out. By default, this field is blank, which tells Compass to use the default implementation. See the Custom Code File examples later in this manual for an example of a custom serializer implementation.

- The **DICOMWEB** value means that Compass will route store requests using the Store Transaction of the Studies Service and Resources protocol (commonly referred to as STOW-RS) of the DICOM PS 3.18 – Web Services specification (commonly referred to as DICOMweb). STOW-RS requests can match both **Store and Forward** or **Direct** rules. If Direct Routing is enabled, Compass will also direct-route query requests using the Query Transaction of the Studies Service and Resources protocol (commonly referred to as QIDO-RS) and the Retrieve Transaction of the Studies Service and Resources protocol (commonly referred to as WADO-RS). QIDO-RS and WADO-RS requests can only match **Direct** rules.

The **edit** link next to this selection allows the user to specify the required parameters for these transactions. There are separate tabs for the **STOW-RS**, **QIDO-RS**, and **WADO-RS** configuration parameters, as well as a separate tab for **Authentication** configuration parameters.

For **STOW-RS**:

The **Resource Path** configures the RESTful service study-level resource path. This is typically the value “/studies”. The **Request Media Type** configures the DICOM media type used for the HTTP request. Compass supports the “application/dicom” media type (which sends instances as binary, DICOM PS 3.10-formatted data), “application/dicom+json” media type (which sends instances as multipart requests containing JSON-encoded metadata, paired with binary-encoded image data), and “application/dicom+xml” media type (which sends instances as multipart requests containing XML-encoded metadata, paired with binary-encoded image data). When configured for “application/dicom”, the **Request Batch Size** configures the maximum number of DICOM instances that will be sent in a single multipart HTTP request. When

configured for either “application/dicom+json” or “application/dicom+xml”, the **Pixel Data URI Encoding** configures the encoding that will be used for the Pixel Data URI, which can be either “As Value” (meaning the URI will be encoded as a Value property/element) or “As BulkData” (meaning the URI will be encoded as a BulkData property/element). And finally, the **Response Media Type** configures the expected media type (i.e., the “Accept” header media type) for the HTTP response. Compass supports the “application/dicom+json”, “application/dicom+xml”, “application/json”, and “application/xml” response media types.

For **QIDO-RS**:

The **Resource Path** configures the RESTful service study-level resource path. This is typically the value “/studies”. The **Response Media Type** configures the expected media type (i.e., the “Accept” header media type) for the HTTP response. Compass supports the “application/dicom+json”, “application/dicom+xml”, “application/json”, “application/xml”, and “multipart/related; type=application/dicom+xml” response media types.

For **WADO-RS**:

The **Resource Path** configures the RESTful service study-level resource path. This is typically the value “/studies”. The **Preferred Transfer Syntax** configures the preferred transfer syntax encoding for the response data set(s). And finally, the **Response Media Type** configures the expected media type (i.e., the “Accept” header media type) for the HTTP response. Compass supports the “application/dicom+json”, “application/dicom+xml”, “application/json”, “application/xml”, and “multipart/related; type=application/dicom+xml” response media types.

For **Authentication**:

If enabled, the **Platform** configures the type of authentication to be performed. Currently, Google Cloud Platform (GCP) is the only supported platform. The **Authentication Method** configures the type of GCP authentication performed. This can be either “Json Key File” or “Certificate”. For “Json Key File”, the **File Path** and **Scope** can be configured. For “Certificate”, the **File Path**, **Password**, **Auth. URL**, **Email**, and **Scope** can be configured. For both types of authentication, the scope is usually the following URL:

<https://www.googleapis.com/auth/cloud-healthcare>

Also note that the Resource Path for GCP usually starts with:

`/v1/projects/{project}/locations/{location}/datasets/{dataset}/dicomStores/{dicomStore}/dicomWeb`

4.2.5.8 DIMSE Relayer

Execute

The user may select “Execute”, which points to a custom implementation for handling such **Direct**-routed messages (by default, the field is blank, which tells Compass to use the default implementation).

Load Balancing

The user may also select “Load Balancing”, which works principally the same as the **Load Balancing Job Action**, except only for **Direct**-routed messages instead of jobs. In this Direct-routed usage, instead of a round-robin selection of which destination will receive a job, there is a

round-robin selection of which destination will receive all **Direct**-routed messages which are received on a given inbound association and routed in **Direct** fashion to a group. See the separate section below on [Load Balancing Job Action](#) for an explanation of the various fields in this configuration.

Query Spanning

This DIMSE Relay will multiplex an inbound C-FIND-RQ message (either MWL or Q/R) across the destinations in the pool. The Compass Destination to which this DIMSE Relay is attached shall be referred to as the **Group Destination**. The **Compass Destinations** in the pool of destinations for round-robin shall be referred to as the Group and/or Group Destinations. The destinations in the pool are queried serially. No order or preference is guaranteed. All intermediate C-FIND-RSP messages will be returned to the original requester. Any final response messages from group members will be dropped by the DIMSE relay. Error messages will not be returned to the original requester but may be logged in the Compass log. The Query Spanning DIMSE Relay will return a generated final C-FIND-RSP with a status of success or failure. Partial failures will not be communicated to the original requester. The Query Spanning DIMSE Relay will not remember any information about what Compass Destinations may be storing any specific DICOM objects.

Destinations in the Group

The selected list of Compass **Destinations** that will be used for Load Balancing or Query Spanning round-robin routing. The **Group Destination** shall not be part of the Group.

4.2.5.9 Use TLS

The **Use TLS** checkboxes specify whether network communications with the specified Destination will be encrypted using TLS (as well as which TLS versions will be supported). The **Ignore Certificate Name Mismatch Errors** and **Allow Self-Signed Server Certificates** checkboxes can be used to relax the Compass certificate validation for this Destination. However, we strongly recommend using these options for testing only, as they greatly reduce security by preventing full TLS authentication from occurring. The **Send Client Certificate** checkbox can be used to control whether or not the client's certificate is sent to the server, enabling client authentication to occur as well (note that server authentication is always mandatory). The default is to allow missing client certificates (no client authentication), which is similar to how web browsers work.

Using the **Send Client Certificate** checkbox also requires that the TLS certificate information be configured correctly in the **System** pane – see the **System** section for more information. The **TLS Certificate** should be set to the location of the certificate that Compass should present for identification to servers. It is suggested that the certificate be a standard PKCS#12 certificate and it must contain an exportable private key. Finally, the **Password** must be set to the password for the private key in the certificate. Using a certificate format that does not password protect the private key allows this setting to be ignored (not recommended for security reasons).

See [Appendix C: Section 1.2 Configuring Secure DICOM Communication](#) for more details about using Compass TLS support.

4.3 Creating a Rule

A new Rule can be created by selecting **Edit > DICOM Options...** from the menu and then selecting the **Rules** pane. Press the green plus button located under the Rules list. This will create a new Rule with the name “New Rule -?” where “?” is the next available number, starting at 0. Rule names are customizable and can be modified at any time by clicking on the name. The list of Rules may be reordered by selecting a Rule and then pressing the up or down arrows directly to the right of the Rules list. When Compass processes the list of Rules for each received image the Rules list is processed from top to bottom and the first match found is used, therefore the ordering of the Rules may be important if potentially more than one could match.



Once a Rule has been created the next step is to configure its Conditions, Selected Destinations, and Rule Options.

4.3.1 Rule Conditions

A Rule’s Conditions determine whether or not its Actions will be applied to an image being processed. A Rule may have multiple conditions, in which case it may be specified that either all the conditions must apply to the image or that only one condition must apply to the image in order for the Rule to match. Another option is that the Rule always matches; effectively declaring it to have no Conditions. Selecting one of three radio buttons, **If ALL Conditions Match**, **If ANY Conditions Match**, and **Always**, will implement one of these three scenarios. Note that if it is desired to have a more complex logic to a Rule’s conditions (e.g. “(A and B) OR (C and D)”), this can be achieved by using the special **All of** or **Any of** condition which are discussed in more detail below.



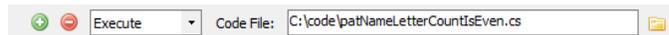
The **Dicom Tag** condition allows incoming images/messages to be selected for this Rule based upon their Dicom Element Tags.



The **Association** condition allows incoming images/messages to be selected for this Rule based upon their incoming Dicom Association information.

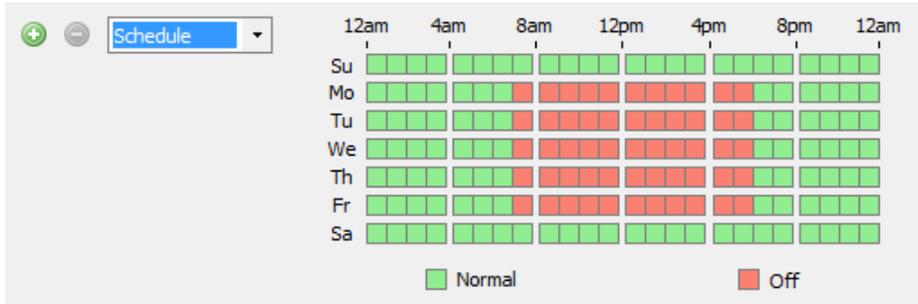


The **Execute** condition allows incoming images/messages to be selected for this Rule based upon a custom, user-defined rule condition. See the example towards the end of this user manual for an example of a custom rule condition implementation.



The **Schedule** condition allows incoming images/messages to be selected for this Rule based upon the time of day and day of week that the message was received. Any squares which are green in the schedule represent time/day blocks where the condition is true; any squares which are red represent time/day blocks where the condition is false. Each square can be changed

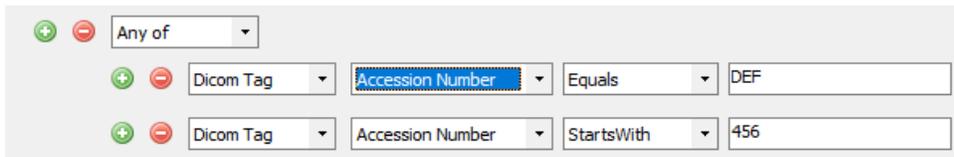
individually by clicking on it. Alternately, the user can right-click to select from a menu of preset schedules for ease of populating the schedule as desired.



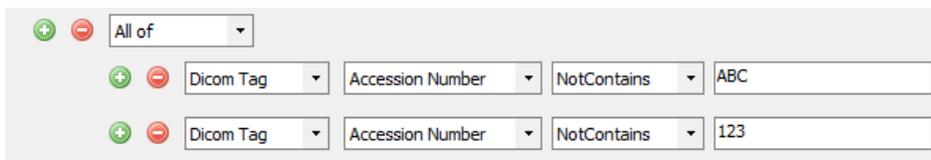
The **Command Tag** condition allows incoming images/messages to be selected based upon information in the DICOM Command portion of the message. This includes information about what type of message (C-STORE-RQ, C-FIND-RQ, etc.) a given message is, as indicated by the content of its Command Field.



The **Any of** condition is a special condition which is true if one or more of its subconditions are true. This condition allows the user to formulate more complex logical Rules.



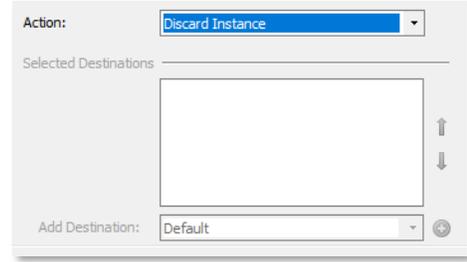
Similarly, the **All of** condition is a special condition which is true if and only if all of its subconditions are true. Again, this condition allows the user to formulate more complex logical Rules.



Configure the condition based upon the desired test. Conditions may be added or removed by pressing the green plus sign or the red minus sign located to the left of each condition.

4.3.2 Rule Actions

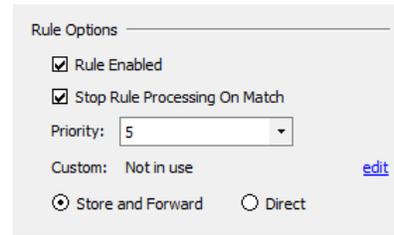
If the logic for the Rule conditions is satisfied, then the image currently being processed will have the currently selected action applied: **Discard Instance** or **Send to Selected Destinations**. To add another Destination to the **Selected Destinations** list, select the desired Destination in the **Add Destination** combo box and press the green plus sign. To remove a Destination from the list, right-click on the desired Destination and choose **Remove Selected** in the context menu. A predefined, special Destination exists called the **Hold Queue**. Any jobs routed to the **Hold Queue** will wait there indefinitely until the user decides to perform an action on them, such as sending them to another Destination or deleting them. Another predefined, special Destination is the **Cache**. Any images routed to the **Cache** will be included in the Cache's Patient/Study/Series/Image hierarchy which is available for view (from the web client) or DICOM Q/R. See the Cache section in this manual for more information.



4.3.3 Rule Options

Uncheck the **Rule Enabled** checkbox to disable the selected Rule. If unchecked, the Rule will be passed over during Rule processing.

Check the **Stop Rule Processing On Match** checkbox if no further Rules in the Rules list should be processed if this Rule matches (Rules are processed sequentially from the top of the list to the bottom of the list). If unchecked, Rule processing will proceed to the next Rule even if this Rule matches.



Specify a number in the **Priority** number chooser to assign the order in which the job that the specified image contains will be processed. 1 is the highest priority, and 10 is the lowest priority. If a job is populated with images that matched different rules containing different priorities, the job is assigned the highest priority of the matching rules. Jobs with a higher priority will be sent before jobs with a lower priority. If a Destination has multiple jobs with the same priority, the older job is processed first.

In some cases, custom code extensions (such as Execute conditions) may want access to configuration data which is added to a particular rule. Click the “edit” link next to **Custom** to see the dialog where this configuration data is provided. In most cases, this data is not needed; when no configuration data has been provided, the rule displays the text “Not In Use”.

Every Rule must be configured to operate in either **Store and Forward** mode or **Direct** mode. The most common (and default) mode of operation is **Store and Forward** mode. In this mode, when an image is received (via DICOM C-STORE-RQ message), if it matches a rule, it will be added to one outbound job for each destination selected in the rule. Compass remembers these jobs, displays them in the Jobs screen, retries them if they fail, etc. Compass also will not start any such outbound job until the incoming association which added images to it is closed (unless the destination is specified in Immediate Mode). **Store and Forward** rules are, of course, allowed to

have more than one destination; additionally, a **Store and Forward** rule may be specified to **Stop Rule Processing on Match** or not. If a Rule is not Stop on Match, then an image may match multiple rules in succession, possibly creating multiple jobs as a result.

Direct mode is fundamentally different. Any DICOM message type (C-STORE-RQ, C-FIND-RQ, etc.) may match a **Direct** mode rule. When such a message matches, Compass will immediately create an outbound association to the destination specified in the rule, forwarding the matching message on to that destination. Compass will also forward any replies from the destination back to the original sender. The nature of this **Direct** routing requires that any **Direct** rule have exactly one destination and that the rule be configured to **Stop Rule Processing on Match**. Additionally, Compass does not remember any messages which match a Direct rule (since they are not added to jobs, and Compass remembers jobs). As a result, they will not appear in the Jobs view and they will not be retried by Compass – responsibility for retrying any failed messages lies with the originator of the message. **Direct** mode routing requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM-routing Compass.

4.4 Special Considerations for Routing WADO-RS (DICOMweb)

Since the DICOMweb transactions STOW-RS and QIDO-RS are fairly similar to their DIMSE counterparts C-STORE and C-FIND respectively, configuring them is fairly straightforward. The DICOMweb transaction WADO-RS, however, is significantly different than its DIMSE counterparts C-MOVE and C-GET (as discussed below) and thus warrants a more detailed explanation.

The DICOMweb WADO-RS transaction is closer to the DIMSE C-GET message in its operation, in that it is a synchronous transaction in which a set of one or more instances (study/series/instance) is queried, and a response is sent back synchronously on the same connection. However, since C-GET is not widely implemented, Compass instead maps WADO-RS into C-MOVE (which is widely implemented) and vice versa. However, due to a functional mismatch between the way WADO-RS works and the way C-MOVE works, configuring Compass to route between the protocols is a bit tricky.

By way of review, a C-MOVE command sent from DICOM Client A to a DICOM Server B instructs that server to C-STORE instances to a second DICOM Server C (which can be optionally collocated with either A or B). As the instances are stored, intermediate status updates are sent back from Server B to Client A. When all instances have been moved to Server C (or failed in the attempt), Server B returns a final status of the move back to Client A.

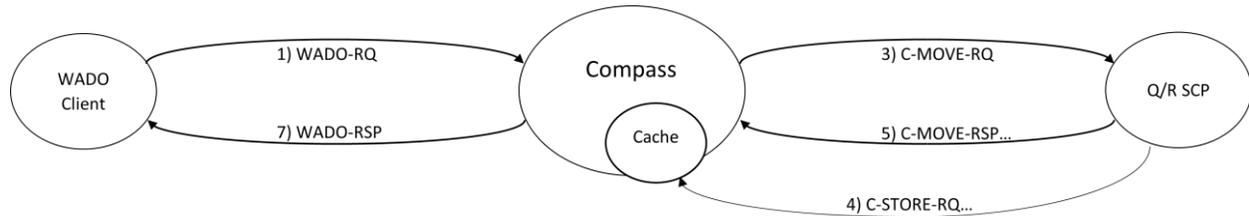
By comparison, a WADO-RS transaction sent from DICOMweb Client A to a DICOMweb Server B instructs that server to send the instances directly back to Client A (in the response). This creates issues when mapping from WADO-RS to C-MOVE and vice versa, as there is no Server C (as discussed in the previous paragraph) involved in the transaction. Compass must therefore assume the role of Server C. The following sections will look at the issues encountered when mapping in each direction, as well as those encountered when mapping from WADO-RS through Compass back to WADO-RS (as might be done when using Compass' filtering capability inline between a WADO-RS client and a WADO-RS server).

4.4.1 Mapping WADO-RS to C-MOVE

When mapping from WADO-RS to C-MOVE, Compass needs to assume the role of the third-party store SCP, since the WADO-RS client has no such capability. Once the instances have been stored to Compass, it can then send these instances back to the WADO-RS client in the response. Compass uses the Compass Cache feature to act as its own private store SCP by first ingesting the instances, using the Cache as a buffer until all instances been stored, and then sending them back to the WADO-RS client. To do this, the following must be configured:

- A DICOMweb source must be configured with a Move AE Title that maps back to the Compass Cache. This Move AE Title will be used as the Move AE for the C-MOVE request that is generated from the incoming WADO-RS request.
- A DICOM (DIMSE) destination must be configured for the Move SCP.
- A Direct Routing rule must be configured to send the generated C-MOVE requests to the above Move SCP.

Below is a diagram and description of the above setup:



Steps:

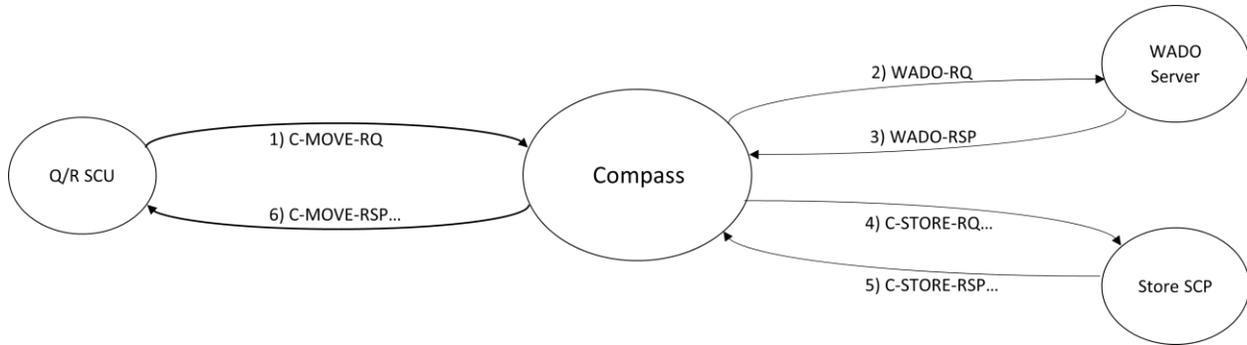
- 1) WADO-RS client sends WADO-RS request to Compass.
- 2) Compass queries for matches in Cache (if found, go to 7).
- 3) Compass sends C-MOVE request to Q/R SCP.
- 4) Q/R SCP sends one or more C-STORE requests to Compass Cache.
- 5) Q/R SCP sends a C-MOVE response for each C-STORE, plus a final response.
- 6) Compass queries for matches in Cache.
- 7) Compass sends WADO-RS response containing matches.

4.4.2 Mapping C-MOVE to WADO-RS

When mapping from C-MOVE to WADO-RS, Compass needs to perform the C-STORES on behalf of the WADO-RS origin server, since WADO-RS has no concept of storing instances to a third party. To do this, the following must be configured:

- A DICOMweb destination must be configured for the WADO-RS server.
- A Direct Routing rule must be configured to send the relevant C-MOVE requests to the above WADO-RS destination.
- A DICOM (DIMSE) destination must be configured with a Move AE Title that matches the Move AE Title from the incoming C-MOVE requests.

Below is a diagram and description of the above setup:



Steps:

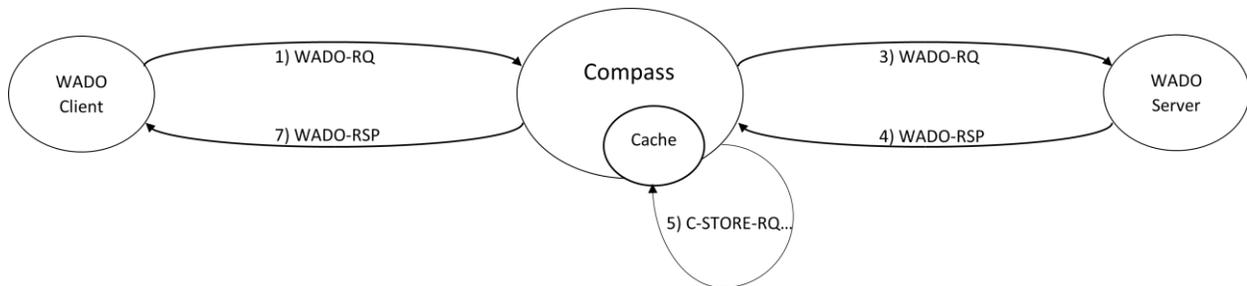
- 1) Q/R SCU sends C-MOVE request to Compass.
- 2) Compass sends WADO-RS request to WADO-RS server.
- 3) WADO-RS server sends WADO-RS response with matches to Compass.
- 4) Compass sends C-STORE requests for each match to Store SCP.
- 5) Store SCP sends C-STORE responses for each request.
- 6) For each C-STORE response, Compass sends a C-MOVE response to Q/R SCU, plus a final response.

4.4.3 Mapping WADO-RS Through Compass Back to WADO-RS

When mapping from WADO-RS through Compass and back to WADO-RS, there is an additional configuration complexity since the DICOM (DIMSE) destination for the outgoing C-MOVE requests must be a loopback (i.e., Compass sending to itself via “localhost”) to the Compass Cache. To do this, the following additional constraint must be met:

- The DICOM (DIMSE) destination from the section above must be a destination that loops back (i.e., sends to itself via “localhost”) to the Compass Cache.

Below is a diagram and description of the above setup:



Steps:

- 1) WADO-RS client sends WADO-RS request to Compass.
- 2) Compass queries for matches in Cache (if found, go to 7).
- 3) Compass sends WADO-RS request to WADO-RS server.
- 4) WADO-RS server sends WADO-RS response with matches to Compass.
- 5) Compass sends C-STORE requests for each match to Compass Cache.
- 6) Compass queries for matches in Cache.
- 7) Compass sends WADO-RS response containing matches.

4.5 Configuring a Job Action

Job Actions can be configured by selecting **Edit > DICOM Options...** from the menu and then selecting the **Job Actions** pane. Press the green plus button located under the Job Actions list. This will create a new Job Action with the name “New Job Action-?” where “?” is the next available number starting with 0. Job Action names are customizable and can be modified at any time by clicking on the name. The list of Job Actions can be reordered by selecting a Job Action and pressing the Up and Down arrows. The ordering of Job Actions in the list reflects their order of execution.

Once a Job Action has been created, the next step is to configure its Trigger, Destinations, and Type.

Trigger

The Trigger can be set to one of the following options:

- **On Sent** – the action will be executed as jobs become marked as Sent
- **On Fail** – the action will be executed as jobs become marked as Failed
- **On Start** – the action will be executed each time a job enters the Running state

Destinations

A Job Action can be associated with jobs for all destinations, or a subset of configured destinations by using the checkboxes to make a selection.

Type

4.5.1 Execute Job Action

Allows custom actions to be defined

Code File

For Execute type actions the location of the file containing the code for the action must be specified. The button with the folder icon can be used to browse to the location of the code file.

Parameters

Execute actions support parameterization. Using the grid, a set of key/value pairs can be defined which will be passed into the action at runtime.

4.5.2 HL7 Send Job Action

Send an HL7 message from a template.

Host

HL7 server hostname

Port

HL7 server listen port

4.5.2.1 HL7 Message

Edit HL7

Open an editor window to create or modify a previously loaded HL7 template.

HL7 Template

A templated HL7 message. This is the HL7 message that will be sent to the server. It can contain variables that will be interpolated. These include a **TIMESTAMP**, a generated Message Id, DICOM elements from the received job, values from key-value pairs added by custom code, and values from the HL7 mapping DB

4.5.2.1.1 Sample HL7 Template Message

```

MSH|^~\&|RAD|CDIC|||${TIMESTAMP(yyyyMMddHHmmss)}||ORM^O01|${GUID()}|P|
2.3
PID||${PATIENT_ID}|CA00116975||TEST^${REGEX(0010,0010
.*\^(.*)})^M||19370702|F|||||||AG0016556211
PV1||O|AGDIDC||| STAFFID^LASTNAME^FIRSTNAME^^^
ORC|NW
OBR||D000603853A||US^ARTEXTCNB^Arterial Extracranial
Bil||201305081047|||||||
STAFFID^LASTNAME^FIRSTNAME^^^|1^^^201305081047^^R|||433.10
Carotid artery stenosis

```

4.5.2.1.2 Variables that can be Interpolated

`${TIMESTAMP(yyyyMMddHHmmss)}`

A timestamp that can be inserted into the outbound HL7 message

`${GUID() }`

A unique identifier that can be used as the Message Id in an HL7 message.

`${DCM(0010,0010)}` or
`${DCM(0010,0010):value_if_missing}`

A DICOM element taken from the first DICOM image in the job. In these cases, 0010,0010, the Patient's Name will be returned if it is present. If Patient's Name is not present, then using the first syntax will result in an error. Using the second syntax, *value_if_missing* will be returned.

`${REGEX(0010,0010 .*\^(.*))}` or
`${REGEX(0010,0010 .*\^(.*)):value_if_missing}`

A C# style regular expression that will be applied to the specified DICOM element from the first image in the job.

Note that there is a space after the tag number (0010,0010 in this example) and also after the actual regular expression. These spaces must be present.

These regex examples will return the first name from the Patient's Name (0010,0010) DICOM element, assuming it is present. If Patient's Name is not present, then using the first syntax will result in an error. Using the second syntax, *value_if_missing* will be returned.

`${USERDATA (user-data-key) }`

Every job maintains a set of key-value pairs. As a job is processed by job actions and filters, custom code can add to and access these key-value pairs. For example, an Execute job action may add a key “accession_number” with a value of “ACC1234” via custom code. A subsequent HL7 Send action may use `${USERDATA (accession_number) }` to interpolate the value “ACC1234” in its place.

Any column header from the HL7 Mapping file used to create the Mapping DB can be used. For example, if one of the headers of the mapping file is `R:PATIENT_ID`, the `${PATIENT_ID}` can be used in the HL7 template. When a row is matched in the Mapping DB, the value of the `PATIENT_ID` column for that row will be returned.

Mapping DB

The mapping DB contains matching values and replacement values that can be interpolated into the HL7 template message.

The mapping DB is created from a Tab delimited mapping file. The first row of the file is the header. Column names will determine whether a column is a matching column or a replacement column. Matching columns shall be prefixed with ‘M:’ and shall have a value of a DICOM element with format `gggg,eeee`. Replacement columns shall be prefixed ‘R:’. The name of a replacement column is the variable name that shall be used in the HL7 message template.

For example, here is a two-line mapping file.

```
M:0010,0020  M:0008,0050  R:PATIENT_NAME
PID1234  ACC5678  DOE^JOHN
```

In this case the first column is Patient ID, the second column is Accession Number. Those are the match columns. In order to have a match on the first row, the incoming DICOM job must have a Patient ID of `PID1234` and an Accession Number of `ACC5678`.

The third column is the replacement column. If the first two columns had a match, then `DOE^JOHN` would replace `${PATIENT_NAME}` in the HL7 template message.

No Match Action

If there is no match in the Mapping DB for a given job

- Continue – Send the DICOM images even though no HL7 message will be sent
- Fail – The job will have a status of Failed and not send the images
- Retry – This will check the Mapping DB again in case it was updated
- Change Destination – Target the job to a different destination, which may have a different job action with a different Mapping DB associated with it.

Reuse HL7 Connection

If true, hold open the HL7 connection after the first HL7 message is sent. The connection will remain open until the job action is reconfigured, the DICOM Output is stopped on the Compass Client, or the remote host receiving HL7 closes the connection.

4.5.2.2 HL7 ACK

Wait For ACK

If checked the job action will wait for an HL7 ACK message.

Fail on AR or AE

This only applies if waiting for ACK messages. If checked, then receiving an AR or AE acknowledgment code will be considered a failure.

ACK Timeout (s)

Number of seconds to wait for the receipt of an ACK

4.5.2.3 Where to look for demographics

The following options are used to determine which DICOM instance from the Compass job to use when interpolating demographic data into the HL7 template.

Images Only

Only look at image objects in the job. Do not look at Structured Reports, GSPS objects, etc. If this feature is enabled non-image objects will not be included when computing the instance number for located demographics.

Instance Number

The images in the job are ordered by the timestamp when Compass received them for the purposes of this feature. Specifying the number “1” would be the first image that was received in the job. If there are twenty DICOM instances in the job, “20” would be the twentieth instance. Specifying a number larger than the total number of instances in the job will use the last DICOM instance in the job. Note that the **Images Only** checkbox determines if non-image objects are used when computing the instance number for this feature. For example, if there are twenty DICOM instances, nineteen images and one Structured Report and **Images Only** is enabled, then “19” would be the maximum instance number that could be specified.

4.5.2.4 On HL7 Message Send Success

Skip Resending HL7 Message on DICOM Job Retry

If checked if the HL7 message is successfully sent it will not be retried if the job is retried due to failures sending the DICOM images in the job.

DICOM Delay (s)

The number of seconds to wait before sending the DICOM job after sending the HL7 message.

Mark Job (Including DICOM images) as Sent

If checked, mark the entire job as Sent on successfully sending the configured HL7 message. No DICOM images will be attempted to be sent (this option is used only when **Trigger** is set to 'On Start').

4.5.2.5 Logging

Logging

Checking this box will log HL7 messages to the Compass application log file.

4.5.3 Load Balancing Job Action

This job action round-robins outbound Compass jobs over a list of configured and available Compass Destinations with optional sticky persistence on a DICOM tag. The Compass Destination to which this job action is attached shall be referred to as the **Group Destination**. Each **Load Balancing Job Action** can only be attached to one Compass **Destination**. The **All Destinations** checkbox cannot be used. Compass jobs will never be routed to the **Group Destination**. It can be configured to point to an invalid IP/Host combination and should not have **Heartbeat Sensing** enabled on it (since it will not receive routed jobs). The **Compass Destinations** in the pool of destinations for round-robin shall be referred to as the Group and/or Group Destinations.

Note: This job action can only be defined with **Trigger Type** of **OnStart**.

Use Persistence File

Use a file for persistence. This requires providing a valid persistence file name.

Persistence File Name

If a file path is provided, then that file shall be used to persist information regarding which Compass **Destination** has received what Compass job and any information for sticky persistence. This rudimentary file-system-based persistence scheme works as follows. Basically, at startup, the `Reinit()` call will load the list of all "seen" jobs from a known disk file. It will then go through that list, purging any "stale" entries, and then write out a new file with the entries it kept. Then, as the Job Action runs, every time it finds a "new" job (i.e. one with an unknown sticky tag value), it adds a row to the end of the file. In this way, the file is always chronological (oldest first). But it is only ever read at startup.

Memory Duration

Number of seconds to remember where a Compass job was routed. The job action will keep a static memory of what jobs it has routed where. The entries in this memory will have a configurable "lifetime", after which they will be purged. Setting this value to zero is equivalent to disabling sticky persistence for this **Load Balancing Job Action**.

Sticky Tag

Either Study Instance UID or Accession Number can be used as a sticky tag. If you decide to change the sticky tag after initial configuration that may temporarily result in cache misses (until the memory duration window has expired) of previously seen studies. Another consideration is that sticky persistence wins over heartbeat failure or success. For example, a job with Study UID 1.2.3.4 gets routed to Destination A. Before the Study UID 1.2.3.4 job has aged off the memory duration window Destination A heartbeat fails. Another job is created with Study UID 1.2.3.4 it will get queued to Destination A and not get sent until Destination A's heartbeat succeeds.

Check Heartbeat

If **Heartbeat Sensing** is being used on a Compass **Destination** in the current group, then factor that into routing decisions. If **Heartbeat Sensing** is enabled, then the **Destination** with heartbeat failure will not be considered eligible for routing. If you are not using **Heartbeat Sensing** than Compass jobs may be queued up to destinations that are presently unavailable.

Logging

If this box is checked and system level logging is enabled than this job action will log debugging messages. Error messages may be posted without this setting being used.

Destinations in the Group

The selected list of Compass **Destinations** that will be used for round-robin routing. The **Group Destination** shall not be part of the Group.

4.5.4 PowerScribe Custom Fields Job Action

This job action requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM routing Compass. Note that this job action may be used for both PowerScribe 360 and PowerScribe One and that references below to PowerScribe 360 also apply to PowerScribe One.

The Compass PS 360 Job Action takes biometric data fields out of a DICOM SR and maps them onto corresponding (via Accession Number) Auto Text templates in PS 360 using the PS 360's web API. A mapping file must be provided to map field names from the DICOM SR into a PS360 report. Optionally, a post processing job action can be provided. The post processing job action allows the modification, addition, or removal of PS360 Custom Fields after the mapping file has been applied to the DICOM SR but before connecting to PS 360. Presently supported are ultrasound SR's such as blood flow rates, fetal ultrasound, kidney, and scans using private GE SR templates such as thyroid exams.

Sample Compass PowerScribe 360 workflow:

- 1) A DICOM exam with an SR is C-Stored to Compass.
- 2) If it is determined if the PowerScribe job action should be used
 - a) Compass uses the Accession Number from the Compass job to query PowerScribe for all possible Custom fields associated with that accession number.
 - i) If the Accession Number is found in PowerScribe go to step 2b
 - ii) If the Accession Number is not found in PowerScribe then the Compass job is marked as Failed and will be retried based on the current Compass configuration.
 - b) Compass will find all possible biometric fields in the SR.
 - c) Compass will use a mapping file to match SR field values to mapped PowerScribe Custom Field names.
 - d) Compass will loop through the Custom field names from PowerScribe for the given Accession Number
 - i) If the current PowerScribe Custom Field name has a mapping from the current DICOM SR Compass will set the value for the custom field in the first Order found for that Accession Number.
 - ii) If there is no match then do nothing.
 - e) The Compass job will be marked as Sent without being forwarded to PACS and will be purged by the Compass job purger.
- 3) If the DICOM exam did not need the PowerScribe Job action it will be marked as Sent and will not forwarded to PACS and will be purged by the Compass job purger.

Server Name

URL to the PowerScribe 360 Web API. For example,
`http://www.SomeURL.com/RadPortal`

User Name

Log in name to the PS 360 Web API

Password

Password to log on to the PS 360 Web API

Site

The site in PS360 name that has the reports that will be updated

SR Field Mapping

Mapping file to match PS360 Custom Fields to values extracted from a DICOM SR.

Post Process

Optional plugin to modify the dictionary of values extracted from the DICOM SR before they are sent to PS360. This plugin can modify, add, or remove Custom Field values.

Skip If Previously Successful

If this job action had succeeded in the past do not rerun on a job retry.

Overwrite Dictated

If checked, attempt to overwrite auto text fields even if the report is in a state that indicates it may have already been dictated. The statuses that will be overwritten are Correction Rejected, Pending Correction, Corrected, Pending Signature, Sign Rejected, and Final. The PowerScribe system may prevent Compass from updating a report with one of those statuses and Compass will show a success status message.

Logging

Enable logging from Job action

Use SR Accession Number

If this option is checked, then use the Accession Number found in the DICOM SR otherwise use the Accession Number from the images in the job.

Mark as Sent

If the job action is successful mark the Compass job as sent without sending the job to its destination.

4.5.4.1 Sample mapping file

Left Ovary Volume	Left_Ovary_Volume	1
Left Ovary Length	Left_Ovary_Length	1
Left Ovary Height	Left_Ovary_Height	1

The mapping file is a tab delimited text file. The first column is a list of space separated search terms. A minus sign '-' can be placed in front of any search term to as a the 'not' modifier. For example, "-date" would disqualify a flattened field name containing "date" from matching. The second column is the mapped PS 360 Custom Field Name. The third column is optional and

contains the number of significant digits for round off. The round off method used is the .NET MidPointRounding.AwayFromZero option. Any value greater than or equal to '5' will be rounded up.

When Compass processes a DICOM SR it flattens the tree of nodes into a table of names constructed from each code value found in the SR tree and the corresponding biometric value. For example: a flattened name from the SR might be **Summary Laterality Left Ovary Volume cm**. The line Left Ovary Volume from the mapping file will match on this name. Therefore, the corresponding value from the SR will be mapped to the PS Custom field name called Left_Ovary_Volume and will be rounded off to 1 significant digit.

4.5.4.2 Post Process API

```
using System.Collections.Generic;

/// <summary>
/// Allows user to plugin a custom mapping function.
/// </summary>
public interface IDataIntegrationPostProcessAction
{
    /// <summary>
    /// Called after the default 1 to 1 Auto text mapping has run.
    /// </summary>
    /// <param name="srDict">Dictionary of values found in DICOM SR</param>
    /// <param name="mappingDict">Dictionary generated from the Mapping File
    /// provided to the job action.</param>
    /// <param name="autoTextDict">Dict of values to send to PS360.</param>
    void Action(Dictionary<string, string> srDict,
        Dictionary<string, MappingDictData> mappingDict,
        Dictionary<string, string> autoTextDict);
}

public struct MappingDictData
{
    public string AutoTextFieldName { get; set; }
    public int? RoundOff { get; set; }
}
```

4.5.4.3 Additional Tools: PowerTools

The Laurel Bridge PowerTools suite includes a tool to scan a DICOM SR file and either create or append to a mapping file to be used by the Compass PowerScribe Custom Fields job action. It can also display the flattened field names contained in the SR as well as suggest Custom Fields names to be used in the PS360 Auto Text templates.

4.5.5 Fluency Job Action

This job action requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM routing Compass.

The Compass Fluency Action takes biometric data fields out of a DICOM SR and maps them onto corresponding Fluency merge fields using an HL7 interface. A mapping file must be

provided to map field names from the DICOM SR into a Fluency report. Optionally, a post processing job action can be provided. The post processing action allows the modification, addition, or removal of Fluency merge fields after the mapping file has been applied to the DICOM SR but before communicating with Fluency. Presently supported are ultrasound SR's such as blood flow rates, fetal ultrasound, kidney, and scans using private GE SR templates such as thyroid exams.

Sample Compass mModal Fluency workflow:

- 1) A DICOM exam with an SR is C-Stored to Compass.
- 2) If it is determined if the Fluency job action should be used
 - a) Compass uses the Accession Number from the Compass job as the Accession Number in the HL7 message generated to Fluency
 - b) Compass will find all possible biometric fields in the SR.
 - c) Compass will use a mapping file to match SR field values to mapped Fluency merge field names.
 - d) Compass will loop through the merge field names that had a match
 - i) If the current Fluency merge field name has a mapping from the current DICOM SR Compass will create an OBX segment with a name/value pair to be sent to Fluency
 - ii) If there is no match then do nothing.
 - e) The HL7 message will be sent to Fluency
 - f) The Compass job will be marked as Sent without being forwarded to PACS and will be purged by the Compass job purger according to the job action configuration.
- 3) If the DICOM exam did not need the Fluency Job action it will be marked as Sent and will not forwarded to PACS and will be purged by the Compass job purger according the job action configuration.

Host

HL7 server hostname

Port

HL7 server listen port

4.5.5.1 HL7 Message

Edit HL7

Open an editor window to create or modify a previously loaded HL7 template.

HL7 Template

A templated HL7 message. This is the HL7 message that will be sent to the server. It can contain variables that will be interpolated. These include a **TIMESTAMP**, a generated Message Id, DICOM elements from the received job, values from key-value pairs added by custom code, and values from the HL7 mapping DB. See the section for the HL7 Send Action for more information on creating HL7 templates.

Here is a sample HL7 template for use with this [Job Action](#)

```
MSH|^~\&|Main st|Compass|||19980814053924||ORU^R01|970814053924670|P|2.3|||NE|NE
PID|||${DCM(0008,0050)}||${DCM(0010,0010)}|
OBR|||${DCM(0008,0050)}^LaurelBridge|||||||||||||||A|||||||
```

SR Field Mapping

Mapping file to match Fluency merge fields to values extracted from a DICOM SR.

Post Process

Optional plugin to modify the dictionary of values extracted from the DICOM SR before they are sent to Fluency. This plugin can modify, add, or remove merge field values.

Logging

Enable logging from Job action

Reuse HL7 Connection

If true, hold open the HL7 connection after the first HL7 message is sent. The connection will remain open until the job action is reconfigured, the DICOM Output is stopped on the Compass Client, or the remote host receiving HL7 closes the connection.

4.5.5.2 HL7 ACK

Wait For ACK

If checked the job action will wait for an HL7 ACK message.

Fail on AR or AE

This only applies if waiting for ACK messages. If checked, then receiving an AR or AE acknowledgment code will be considered a failure.

ACK Timeout (s)

Number of seconds to wait for the receipt of an ACK

4.5.5.3 Where to look for demographics

The following options are used to determine which DICOM instance from the Compass job to use when interpolating demographic data into the HL7 template.

Use SR Accession Number

If this option is checked, then use the Accession Number found in the DICOM SR otherwise use the Accession Number from the images in the job.

Continue without SR

If this option is checked, then skip sending information to Fluency but allow the Compass DICOM job to continue.

Skip Resending HL7 Message on DICOM Job Retry

If checked if the HL7 message is successfully sent it will not be retried if the job is retried due to failures sending the DICOM images in the job.

DICOM Delay (s)

The number of seconds to wait before sending the DICOM job after sending the HL7 message.

Mark Job (Including DICOM images) as Sent

If checked, mark the entire job as Sent on successfully sending the configured HL7 message. No DICOM images will be attempted to be sent (this option is used only when **Trigger** is set to 'On Start').

4.5.5.4 Sample mapping file

```
Left Ovary Volume    Left_Ovary_Volume    1
Left Ovary Length    Left_Ovary_Length    1
Left Ovary Height    Left_Ovary_Height    1
```

The mapping file is a tab delimited text file. The first column is a list of space separated search terms. A minus sign '-' can be placed in front of any search term to as a the 'not' modifier. For example, "-date" would disqualify a flattened field name containing "date" from matching. The second column is the mapped PS 360 Custom Field Name. The third column is optional and contains the number of significant digits for round off. The round off method used is the .NET MidPointRounding.AwayFromZero option. Any value ending in '5' will be rounded up.

When Compass processes a DICOM SR it flattens the tree of nodes into a table of names constructed from each code value found in the SR tree and the corresponding biometric value. For example: a flattened name from the SR might be **Summary Laterality Left Ovary Volume cm**. The line `Left Ovary Volume` from the mapping file will match on this name. Therefore, the corresponding value from the SR will be mapped to the PS Custom field name called `Left_Ovary_Volume` and will be rounded off to 1 significant digit.

4.5.5.5 Post Process API

```
using System.Collections.Generic;

/// <summary>
/// Allows user to plugin a custom mapping function.
/// </summary>
public interface IDataIntegrationProcessAction
{
    /// <summary>
    /// Called after the default 1 to 1 Auto text mapping has run.
    /// </summary>
    /// <param name="srDict">Dictionary of values found in DICOM SR</param>
    /// <param name="mappingDict">Dictionary generated from the Mapping File
    /// provided to the job action.</param>
    /// <param name="autoTextDict">Dicy of values to send to PS360.</param>
    void Action(Dictionary<string, string> srDict,
        Dictionary<string, MappingDictData> mappingDict,
        Dictionary<string, string> autoTextDict);
}

public struct MappingDictData
{
    public string AutoTextFieldName { get; set; }
    public int? RoundOff { get; set; }
}
```

4.5.5.6 Additional Tools: PowerTools

The Laurel Bridge PowerTools suite includes a tool to scan a DICOM SR file and either create or append to a mapping file to be used by the Compass Fluency job action. It can also display the flattened field names contained in the SR as well as suggest merge field names to be used in the Fluency report templates.

4.5.6 Allscripts Job Action

This job action requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM routing Compass.

The Compass Allscripts Send Action takes DICOM images from a Compass job, converts them to PDF documents, and then encapsulates them in an HL7 message that can be received by an Allscripts HL7 interface. Only image objects are supported at this time. For cine loops, only the first image in the multi-frame will be converted.

4.5.6.1 Configuration

Host

HL7 server hostname

Port

HL7 server listen port

4.5.6.2 HL7 Message

Reuse HL7 Connection

If true, hold open the HL7 connection after the first HL7 message is sent. The connection will remain open until the job action is reconfigured, the DICOM Output is stopped on the Compass Client, or the remote host receiving HL7 closes the connection.

Edit HL7

Open an editor window to create or modify a previously loaded HL7 template.

HL7 Template

A templated HL7 message. It should not include any OBX segments. The OBX segments will be auto generated by Compass. This is the HL7 message that will be sent to the server. It can contain variables that will be interpolated. These include a **TIMESTAMP**, a generated Message Id, DICOM elements from the received job, values from key-value pairs added by custom code, and values from the HL7 mapping DB. See the section for the HL7 Send Action for more information on creating HL7 templates.

4.5.6.3 HL7 ACK

Wait For ACK

If checked the job action will wait for an HL7 ACK message.

ACK Timeout (s)

Number of seconds to wait for the receipt of an ACK

Fail on AR or AE

This only applies if waiting for ACK messages. If checked, then receiving an AR or AE acknowledgment code will be considered a failure.

4.5.6.4 Where to look for demographics

The following options are used to determine which DICOM instance from the Compass job to use when interpolating demographic data into the HL7 template.

Images only

If checked, only images are used to look for demographics data.

Instance Number

The images in the job are ordered by the timestamp when Compass received them for the purposes of this feature. Specifying the number “1” would be the first image that was received in the job. If there are twenty DICOM instances in the job, “20” would be the twentieth instance. Specifying a number larger than the total number of instances in the job will use the last DICOM instance in the job. Note that the **Images Only** checkbox determines if non-image objects are used when computing the instance number for this feature. For example, if there are twenty DICOM instances, nineteen images and one Structured Report and **Images Only** is enabled, then “19” would be the maximum instance number that could be specified.

4.5.6.5 On HL7 Message Send Success

Skip Resending HL7 Message on DICOM Job Retry

If checked, an HL7 message that was successfully sent via a previous attempt will not be resent when the DICOM images are retried.

Mark Job (Including DICOM images) as Sent

If checked, mark the entire job as Sent on successfully sending the configured HL7 message. No DICOM images will be attempted to be sent (this option is used only when **Trigger** is set to 'On Start').

Delay DICOM Job (s)

The number of seconds to wait before sending the DICOM job after sending the HL7 message.

4.5.6.6 Logging

Logging Enabled

If checked, logging will be enabled for this job action.

4.6 Creating a Listener

A Listener is a server-side socket waiting to accept DICOM association requests. Compass contains two predefined Listeners: **UnencryptedDefault** and **EncryptedDefault**. If the current version of Compass was upgraded from a version prior to the existence of the Listeners feature, the **UnencryptedDefault** listener will contain the settings from the prior unencrypted port configuration, and the **EncryptedDefault** listener will contain the settings from the prior encrypted port configuration.

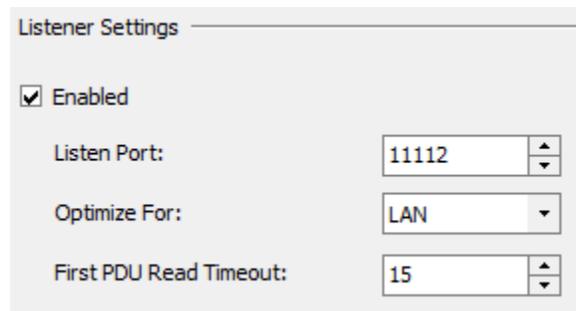
A new Listener can be created by selecting **Edit > DICOM Options...** from the menu and then selecting the **Listeners** pane. Press the green plus button located under the Listeners list. This will create a new Listener with the name “New Listener -?” where “?” is the next available number, starting at 0. Listener names are customizable and can be modified at any time by clicking on the name. The list of Listeners may be reordered by selecting a Listener and then pressing the up or down arrows directly below the Listeners list. Unlike Sources, there is no priority associated with Listener ordering; it is merely provided as a convenience. Numerous options for managing the Listeners list are also available by right-clicking the Listeners pane and selecting an option from the context menu.

Once a Listener has been created the next step is to configure its Listener Settings and Encryption Settings.

4.6.1 Listener Settings

A listener may be enabled or disabled by using the **Enabled** checkbox. At least one Listeners must be enabled. The **Listen Port** number to listen for association requests must be unique.

The **Optimize For** chooser will direct Compass to optimize its network buffers for your particular network topology. If many of your Sources connect to Compass via the Internet it is recommended that you choose WAN for this setting. Otherwise, the default setting of LAN is appropriate.



The screenshot shows the 'Listener Settings' dialog box. It contains the following elements:

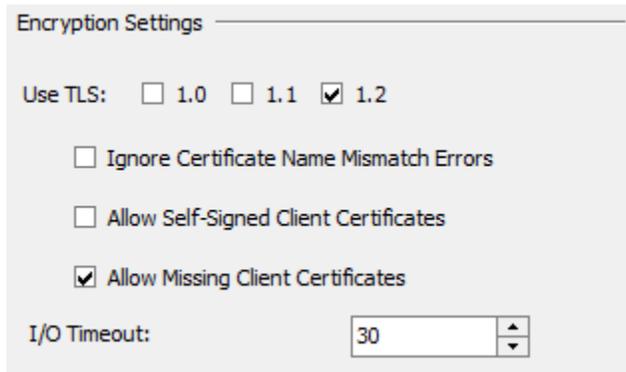
- An 'Enabled' checkbox, which is checked.
- A 'Listen Port' field with a text input containing '11112' and a spinner control to its right.
- An 'Optimize For' dropdown menu currently set to 'LAN'.
- A 'First PDU Read Timeout' field with a text input containing '15' and a spinner control to its right.

After the listener accepts a socket connection, if no DICOM association request or other DICOM data is received within **First PDU Read Timeout** seconds, then the connection is closed.

4.6.2 Encryption Settings

Under Encryption Settings, configurable options include which protocols to allow (if any), certificate options, as well as I/O and authentication timeout options.

Selecting at least one of the **Use TLS** checkboxes makes this an encrypted Listener. Only the TLS versions corresponding to the selected checkboxes will be supported.



TLS

The **Ignore Certificate Name Mismatch Errors** and **Allow Self-Signed Server Certificates** checkboxes can be used to relax the Compass certificate validation for these connections. However, we strongly recommend using these options for testing only, as they greatly reduce security by preventing full TLS authentication from occurring. The **Allow Missing Client Certificates** checkbox can be used to control whether or not client certificates (i.e., client authentication) are required (note that server authentication is always mandatory). The default is to allow missing client certificates (no client authentication), which is similar to how web browsers work.

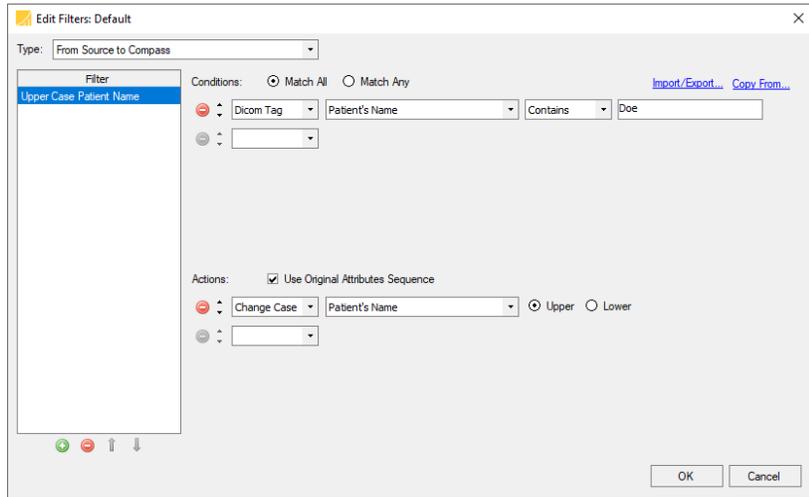
Using TLS also requires that the TLS certificate information be configured correctly in the **System** pane – see the **System** section for more information. The **TLS Certificate** should be set to the location of the certificate that Compass should present for identification to clients. It is suggested that the certificate be a standard PKCS#12 certificate and it must contain an exportable private key. Finally, the **Password** must be set to the password for the private key in the certificate. Using a certificate format that does not password protect the private key allows this setting to be ignored (not recommended for security reasons).

See [Appendix C: Section 1.2 Configuring Secure DICOM Communication](#) for more details about using Compass TLS support.

4.7 Filters

Filters allow a message's header data to be changed as Compass receives it from a Source or as Compass sends it out to a Destination. Additionally, Filters can change response messages received from a Destination (in **Direct** mode) or sent back to a Source. **Actions** are only applied if the **Conditions** are met.

A new Filter can be created by selecting **Edit > DICOM Options...** from the menu, selecting a Source or Destination, and then selecting the **Filters edit** link. The Type selector at the top allows choice between the Filter direction (to or from Compass) and defaults to the usual choice ("Source to Compass" for Sources and "Compass to Destination" for Destinations). Press the



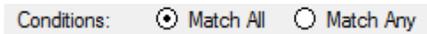
green plus button located under the Filters list. This will create a new Filter with the name "New Filter ?" where "?" is the next available number, starting at 0. Filter names are customizable and can be modified at any time by clicking on the name. The list of Filters may be reordered by selecting a Filter and then pressing the up or down arrows. Filters are applied in order, top to bottom, so ordering is important.

Another way to create a new Filter is to copy from another Source or Destination. Pressing the "Copy From..." link in the upper-right section of the Filter Editor allows you to copy individual Filters defined in the other Sources and Destinations. There is also an option to copy all the Filters from a specific Source or Destination by selecting the menu item named "All" listed below the defined Filters. If a Filter already exists with the same name when copying a Filter, the new Filter will have an asterisk (*) appended to its name.

Once a Filter has been created the next step is to configure its Conditions and Actions.

4.7.1 Conditions

A Filter's Conditions determine whether or not its Actions will be applied to the image being processed. A Filter may have multiple conditions, therefore an option is provided that all Conditions must match or any Conditions can match in order for the Filter to be applied. Another configuration is to have no Conditions, thereby always matching.



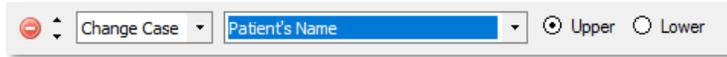
For each Condition select the **Dicom Tag** condition type (additional types may be added in the future). Configure the rest of the Condition based upon the desired test. Conditions may be

removed by pressing the red minus sign located to the left of each condition and sorted via the up and down arrow buttons.

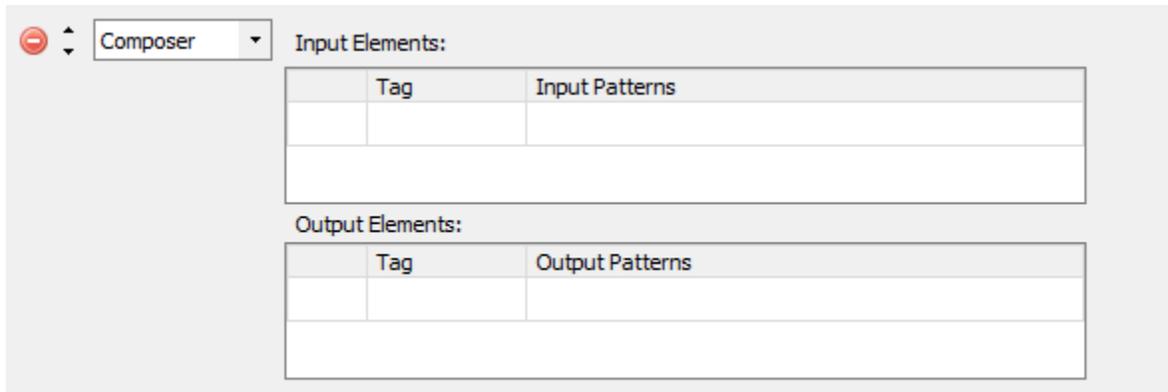
4.7.2 Actions

If the Conditions testing was satisfied, then any defined Actions will be applied to the image.

The **Change Case** action changes the case of the selected tag to either upper-case or lower-case.



The **Composer** action supports the merging and splitting of elements. It can also be used to manipulate DICOM sequence elements with ease. It uses regular expressions to parse values from DICOM tags and combine the values into other DICOM tags. For example, it can take parts from two different tags and combine them to make a new value in a third tag. Regular expressions are specified to parse each input tag and substitution patterns determine how the output tags are constructed.



In the “Input Elements” table, specify the tags and how the regular expressions should parse each value into groups – the regular expressions go into the “Input Patterns” column. The groups that are produced are specified in the Output Patterns column of the “Output Elements” table. The first match in the first input element is referred to as “\${1.1}”, the second match in the first input element is “\${1.2}”, and the third match in the second input element would be “\${2.3}”, and so on.

The Output Elements specify what parts of the input elements to combine and how to combine them; note that the parts can be used multiple times and also combine them with plain text.

The **Copy** action copies the tag specified in the **From** field to the tag specified in the **To** field.



The **Execute** action executes a piece of custom code that adheres to a particular interface:

```
using LaurelBridge.DCF.Configuration;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.Filters;

public class DateComparisonFilter : IFilterAction
{
    public void ApplyAction(CFGGroup config, DicomSessionSettings dss,
        RelevantTagMarker tagMarker, ref DicomDataSet dds)
    {
        // custom code
    }
}
```



The **Find/Replace** action changes the value of the selected tag via hardcoded values or regex values.

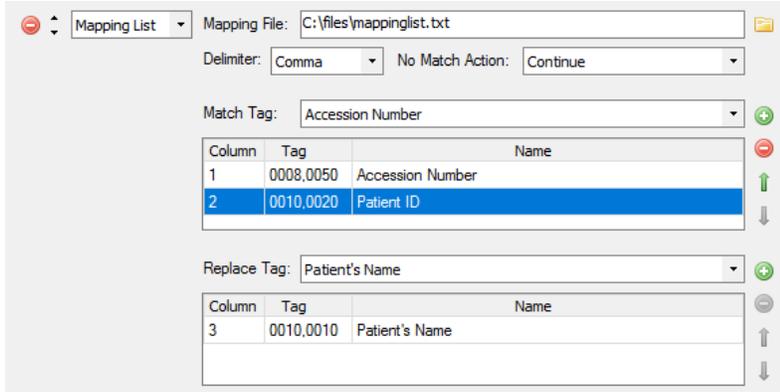


The **Insert/Overwrite** action creates the specified **Tag** (if it didn't already exist) or overwrites the specified tag (if it did already exist) with the specified **Value**.



The **Mapping List** action allows the user to specify an external file that lists match tags and replace tags. When creating a Mapping List action in Compass, the user specifies the format of the mapping list file. The example below shows that the mapping list file will contain three comma-separated fields: an accession number, a patient ID, and a patient's name. If the match tags on any given row match the current dataset, the values of the replace tags are substituted into the dataset. Below is an example of a mapping list file that matches the format of the mapping list action:

```
PIKR0004, PAT124, SMITH^JOSEPH
ABC123, PAT45, MINER^STEVEN
```



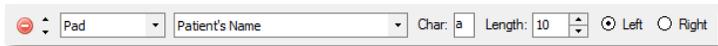
The **Move** action moves the tag from its original location to its final location. Once moved, the tag will not be present in its original location.



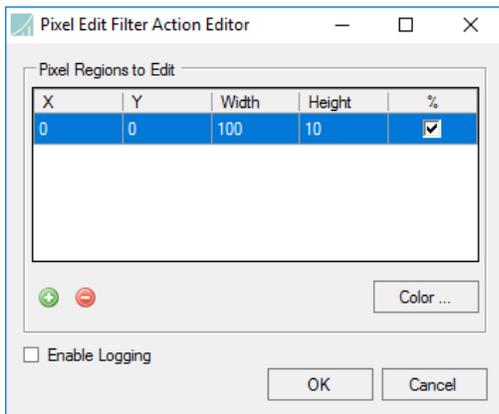
The **New UID** action replaces the Unique Identification (UID) value of the specified UID type tag in the **Tag** field with a new UID value.



The **Pad** action pads the specified tag with the specified character. The **Left** and **Right** radio buttons specify whether to pad the left or right side of the value. The new total length of the tag value is specified by the **Length** field. If the length of the original value is equal to or greater than the Length field, the value will not be changed.



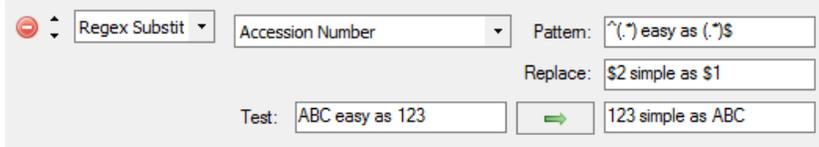
The **Pixel Edit** action allows modification of regions of the pixel data. For more information on the configuration of this **Action**, see the section below under **De-Identify Filter Action**.



The **Prepend/Append** action prepends or appends the specified text to the specified tag's value.



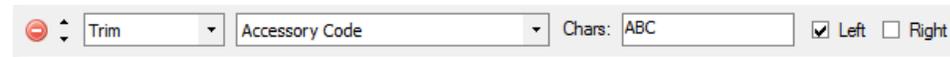
The **Regex Substitution** action performs a regular expression match on the specified tag’s value using the given **Pattern**. It then replaces the value with the **Replace** string which may contain regex variables from the match. The **Test** box can be filled with an example tag value and clicking the green arrow processes the test data.



The **Remove** action removes the specified tag. If the **Only if empty** checkbox is checked, the tag will be removed only if the specified tag’s value is empty.



The **Trim** action removes the specified individual characters from the left, right, or both sides of the specified tag’s value. Please note that the characters are a list of individual characters, and not a phrase treated as a whole. For example, if the characters listed in the Chars text box are “ABC”, and the “Left” checkbox is checked, an input string of “BACAZZHELB” would be trimmed to “ZZHELB”. The “B” matches and is trimmed. Then the “A” matches and is trimmed. Then the “C” matches and is trimmed. Then the “A” matches and is trimmed. The “Z” does not match and trimming stops.



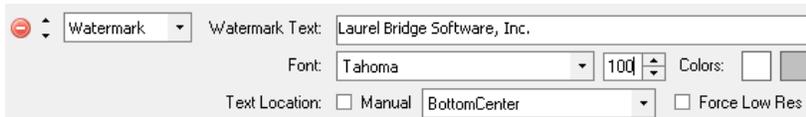
4.7.3 Original Attribute Sequence

The original tag value of a filtered tag can optionally be stored in the original attribute sequence by checking the **Create Original Attribute Sequence** checkbox. If left unchecked, the original values modified by the filter will not be recorded in the original attribute sequence.



4.7.4 Watermark

The **Watermark** action embeds customized text within an image at a predefined location within the image. Two separate modes are provided for indicating the placement of the watermark, Automatic and Manual.

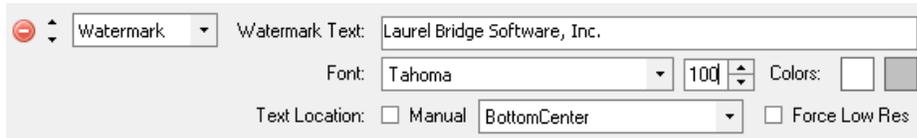


4.7.4.1 Options

The list of options available to configure the watermark filter are as follows:

- **Watermark Text:** This is the actual watermark text. Keep in mind the watermark will scale to fit within the region selected, so it is best to keep this text terse. For longer text it may be better to use Manual mode, as more of the image region is utilized.
- **Font Name:** The name of the font family to use to render the watermark text in higher resolution images. (Force Low Res is unchecked). When the resolution of the watermark text falls below a certain threshold where the text will no longer be readable, this font will automatically switch to a bitmap pixel font that is more readable at smaller font sizes.
- **Font Size:** The size of the watermark text to attempt. If the watermark text at this designated size does not fit, the font will automatically be scaled down to fit the region specified.
- **Primary Font Color:** The primary font color to use for the watermark text. This will be the entire text color when low resolution processing is enabled.
- **Secondary Font Color:** The secondary font color to use for the watermark text. This is the border color for each character in the watermark text. This color is ignored when low resolution processing is enabled.
- **Text Location Manual:** The manual check box enables manual mode, which allows the user to specify the exact region to place the upper left corner of the watermark text. The remaining width of the image will be used as the region to place the watermark text.
- **Text Location Automatic Region:** One of the available regions within the image to attempt to place the watermark text. The watermark text will be automatically scaled down to fit within this region by decreasing the font size.
- **Force Low Res (Resolution):** If enabled, low resolution processing is enabled which:
 - Forces the font to be a pixel font, which performs better at lower resolutions
 - Renders the watermark text more accurately for lower resolutions
 - Disables the secondary font color
 - This option can potentially introduce textual artifacts, such as aliasing and jagged edges, as a trade-off for readability. This option should only be enabled for smaller image sizes where the normal watermark process produces unreadable text.

4.7.4.2 Automatic Watermark Text Placement



The image above demonstrates the options available when configuring the watermark filter in automatic mode.

Automatic watermark text placement provides the user with the following choices for text placement, where the outer rectangle represents the image, and each selection represents a region within the image where the watermark is to be placed:

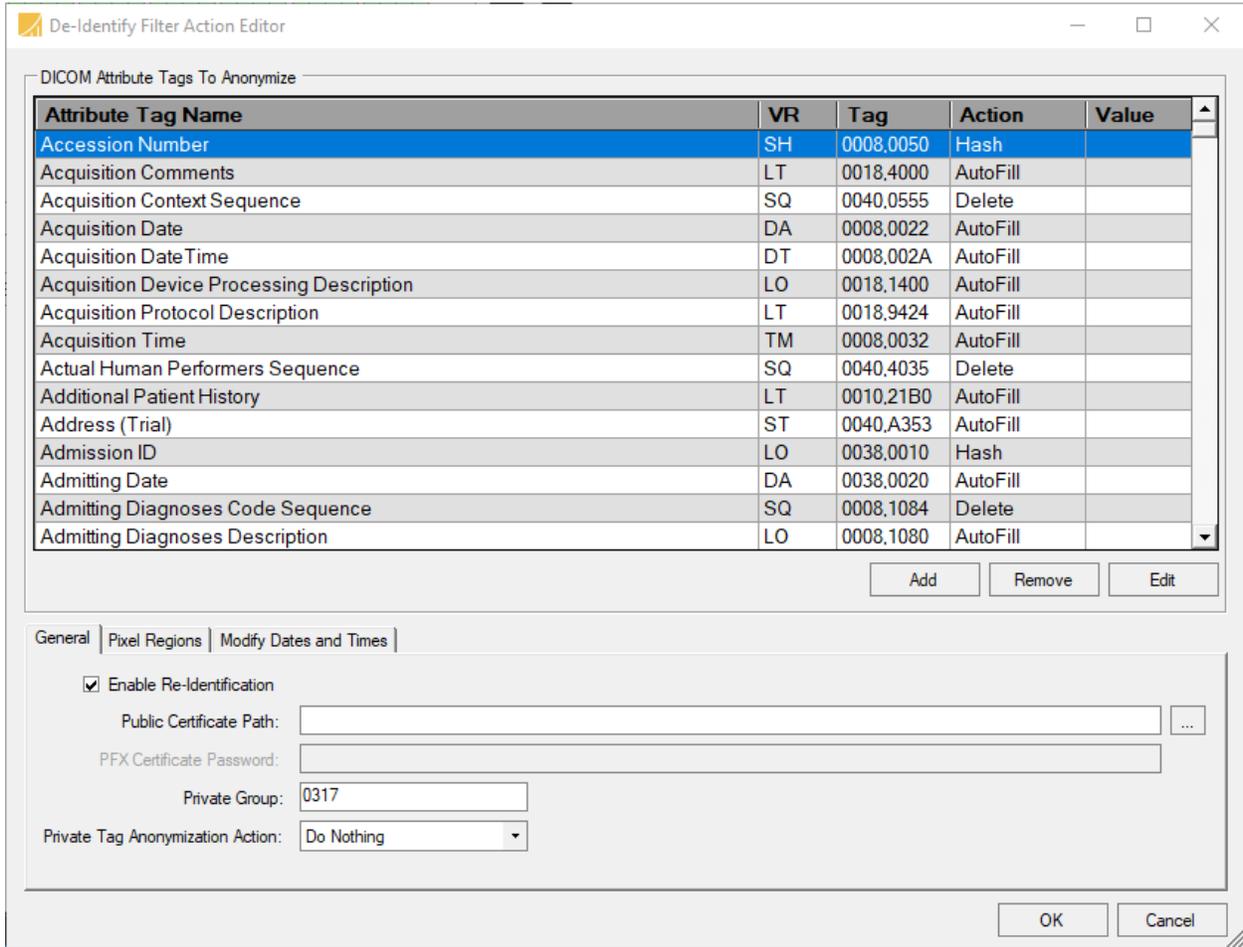
TOP LEFT	TOP CENTER	TOP RIGHT
MIDDLE LEFT	MIDDLE CENTER	MIDDLE RIGHT
BOTTOM LEFT	BOTTOM CENTER	BOTTOM RIGHT

4.7.4.3 Manual Watermark Text Placement

Watermark Text: Laurel Bridge Software, Inc.
Font: Tahoma 100 Colors: [] []
Text Location: Manual X: 0 Y: 0 Force Low Res

When the ‘Manual’ checkbox is enabled, the X and Y numeric up/down controls appear to allow the user to specify a location within the image to place the top left corner of the watermark text. The X coordinate indicates the column location, and the Y coordinate indicates the row location. The remaining width (image columns – x) is used as available space when determining if the watermark text can be rendered at the specified font size.

4.7.5 De-Identify Filter Action



The **De-Identify Filter Action** provides a mechanism to remove and replace certain attributes within a DICOM dataset that may lead to patient identification.

4.7.5.1 DICOM Attribute Tags to Anonymize

By default, this list is populated with attributes suggested in **PS 3.15 Annex E Attribute Confidentiality Profile** in the 2011 DICOM specification. Attribute tags may be added to the list of DICOM tags to anonymize by using the ‘Attribute Tag’ drop-down combo box, which contains an enumeration of the public tags found in the 2011 DICOM dictionary. Public tags not found in the dictionary, nor private tags, may be added to the list of attribute tags to anonymize using this control. **See Section 4.7.5.5** for additional information regarding how the filter De-Identifies private tags. The following describes each aspect of an entry in the data grid shown above under **DICOM Attribute Tags to Anonymize**:

- **Attribute Tag Name** – Specifies the name of the attribute tag to be anonymized if present in the DICOM dataset’s header data.
- **VR** – Indicates the attribute’s Value Representation. This becomes useful when determining an appropriate replacement value for de-identification when “AutoFill” is selected.
- **Tag** – Indicates the attribute tag of the attribute to be anonymized.

- **Action** – Specifies the replacement action to use for a given attribute when determining the replacement value. The available replacement actions are determined based on the attribute’s VR. The following list contains all available replacement actions:
 - **Hash (SH, LO, PN elements)** – For SH, LO, and PN elements, the replacement value will be a trimmed hash string (using SHA256) of the original value.
 - **GenerateUID** – For a given UI element, a new UID will be generated for the replacement value.
 - **TransformUID** – For a UI element, the existing UID will be transformed in a consistent manner to allow for repeatability. For instance, this option is typically selected for SOP Instance UIDs to allow a given UID to map to the same new UID across multiple instances of the **De-Identify Filter Action**.
 - **AutoFill** – Replace the current DICOM attribute value with an appropriate replacement value based on the attribute’s VR. The replacement value is predetermined and is not configurable.
 - **UserDefined** – Replace the current attribute value with the given replacement string defined in the “Value” column.
 - **Default** – Replace the current DICOM attribute value with a user-specified value. For example, specifying the default of ‘M’ for Patient’s Sex will set all instances of Patient’s Sex to ‘M’ by default.
 - **Delete** – Specifying this option will remove the given attribute from the de-identified dataset. If “Delete” is selected for a given attribute and the attribute is present in a sequence (SQ), that sequence will be removed from the de-identified dataset.
- **Value** – If “UserDefined” is selected as a replacement value action, the string specified in the Value column will be used as a replacement value for the given attribute. This value will appear as is. Care should be taken when entering a value for non-standard string VR types (“CS” for instance).
- **Delete** – Press the delete button to remove the given attribute from the list of attributes to de-identify.

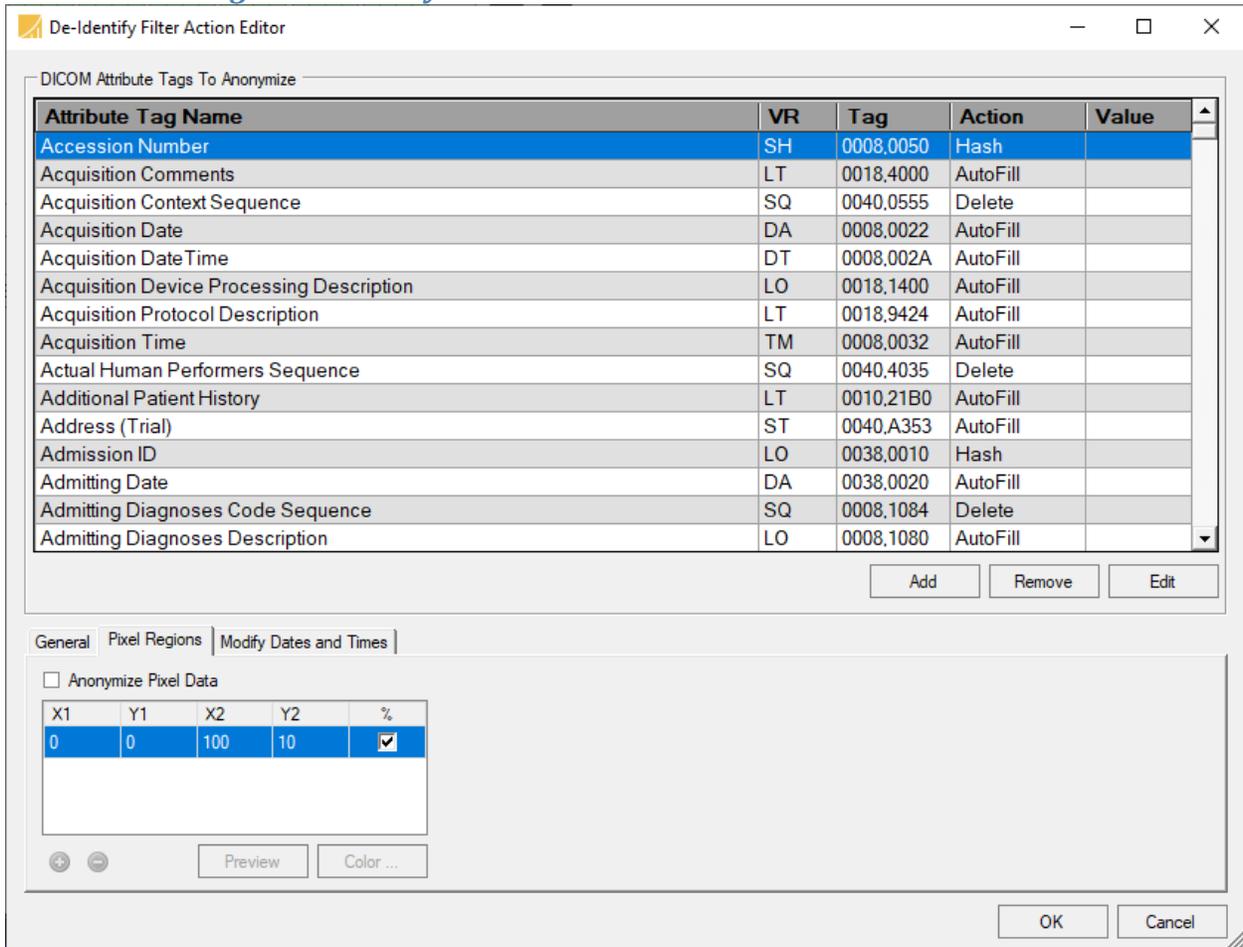
4.7.5.2 General Settings

These settings allow for finer-grained control over common options needed during de-identification.

- **Enable Re-Identification** – If checked, the original values for each attribute anonymized will be persisted in the resulting de-identified dataset using the specified public certificate. More information about this process is specified later in this section.
- **Public Certificate Path** – If Enable Re-Identification is checked, a public certificate, given in the binary DER format as a .crt, is required to properly encrypt the original PHI for later Re-Identification. The PFX certificate, which contains the public certificate and private keys, is also supported.
- **PFX Certificate Password** – If Enable Re-Identification is checked and a PFX certificate is specified in the Public Certificate Path, the password for the certificate’s private keys must be specified.
- **Private Group** – The Private Group box specifies which group will be used during the de-identification process. After De-Identification, this private group will hold the Encrypted Attribute Sequence containing the original data.

- **Private Tag Anonymization Action** – Setting to removing all private tags from the DICOM dataset’s header data. The current options and their descriptions are as follows:
 - **Do Nothing** – Leave the private tags as they are. By default, this option is selected.
 - **Remove All** – Remove all private tags from the DICOM dataset’s header data. This does not include any private tags added during the de-identification process. This removal is unidirectional and cannot be undone, even during re-identification.
 - **Encrypt All** – Remove all private tags from the DICOM dataset’s header data. This does not include any private tags added during the de-identification process. This option can only be enabled if **Enable Re-Identification** is selected, and a valid public certificate is provided.

4.7.5.3 Pixel Regions to Anonymize



The Pixel Regions tab shown above contains configuration items that pertain directly to anonymizing pixel regions from within a data set.

- **Anonymize Pixel Data** – If selected, the list of rectangles defined in the **Pixel Regions to Anonymize** grid will be removed from the DICOM dataset’s pixel data. This includes pixel data in (7FE0, 0010), and not any additional overlays or private tags. This option will be skipped if no valid pixel data is present in the DICOM dataset.

This grid specifies which rectangles to anonymize in a DICOM dataset's pixel data (7FE0,0010). Each region is defined by providing the two corner points of a rectangle, (x1,y1) and (x2,y2), as either a percentage or absolute coordinates. The following section will outline the various options, as well as provide simple examples to demonstrate the various configuration settings.

The indicated regions to anonymize are only anonymized if the 'Anonymize Pixel Data' option is checked from the Misc. Settings section of the De-Identify Filter Action Editor. These same pixel regions are restored during re-identification if the 'Enable Re-Identification' option is selected, and a valid Public Certificate is provided.

Percentage

Both points (x1,y1) and (x2,y2) may be given as a percentage of the image's rows (y) and columns (x). This option is the most flexible, as it does not require prior knowledge of the image size of the DICOM image to anonymize.

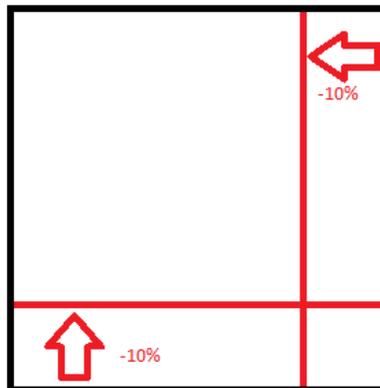
For instance, the following images demonstrate the sample configuration for blanking out the top and bottom 10% of an image:

Consider the following percentage calculations for an 800 x 600 image:

```
Image Size: 800w (columns) x 600h (rows)
Region 1: x1, y1 = (0%, 0%) x2, y2 = (100%, 10%)
          x1, y1 = (0%x800=0, 0%x600=0)
          x2, y2 = (100%x800=800, 10%x600=60)
Region 2: x1, y1 = (0%, -10%) x2, y2 = (100%, 100%)
          x1, y1 = (0%x800=0, -10%x600=540)
          x2, y2 = (100%x800=800, 100%x600=600)
```

Negative Pixel Region Coordinates

Note how the above region 2 was calculated with a y1 percentage of -10%. Negative percentages correlate to percentages starting from the other end of image. In other words, negative percentages for width mean starting from the right side of the image opposed to the left. See the illustration below for more information about negative coordinate values. Note that negative values may be applied to both percentages and absolute coordinates and are treated the same.



Negative Pixel Region Coordinates

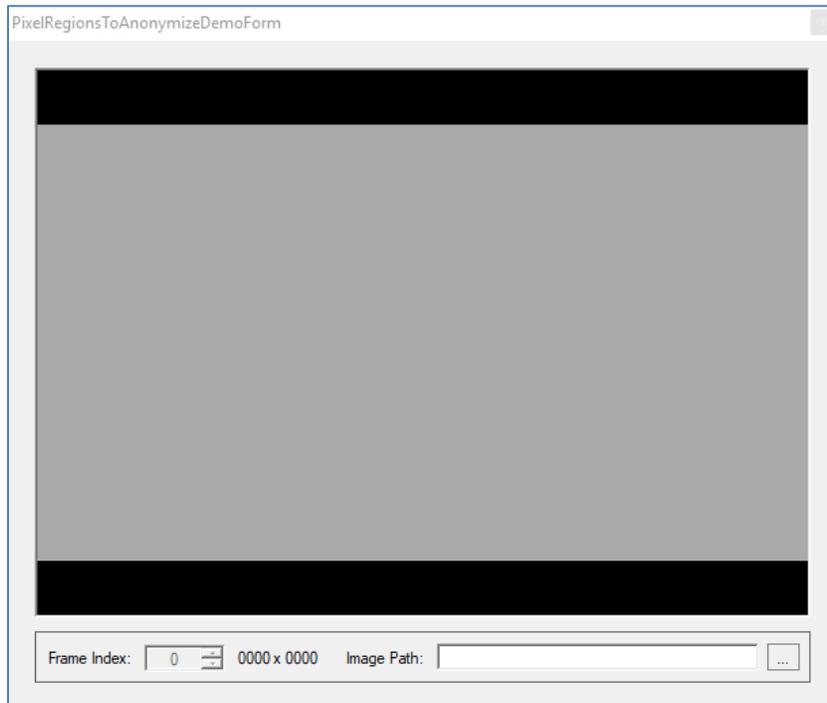
Absolute Coordinates

Both coordinates (x1, y1) and (x2, y2) may also be given in absolute coordinates in reference to the actual number of rows and columns for a particular image. This mode is only recommended if the type of DICOM image remains constant for this filter, and the number of rows and columns are known at this point. This mode tends to be more precise than percentages when the exact location of the region (or regions) to anonymize are already known.

Demo Button

A demonstration button has been added to the Pixel Regions to Anonymize group to assist in the configuration of the list of regions to anonymize. After configuring the data grid with one or more regions to anonymize (again by specifying (x1, y1) and (x2, y2) coordinates), press the 'Demo' button to view a thumbnail bitmap of the regions specified to anonymize. (Please refer to the [Pixel Regions To Anonymize Demo Form](#) image below for an example). The Pixel Replacement Color selected in the previous form will be used to render the rectangles seen drawn on the thumbnail bitmap.

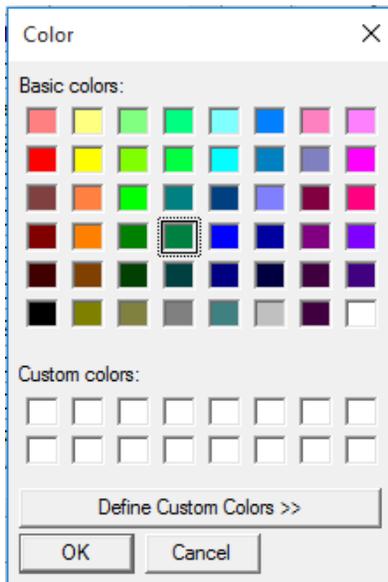
NOTE: At the point the demonstration form is launched, the actual image rows and columns are not known. This will lead to inaccurate renderings when using absolute coordinates until an actual image is loaded via the '...' button located by the image path text box.



Pixel Regions to Anonymize Demo Form

Color Button

Clicking **Color...** allows the user to define an appropriate replacement pixel value for each type of supported photometric interpretation via the following dialog box:



De-identify pixel region color picker

The supported photometric interpretations for de-identification are:

- MONOCHROME 1
- MONOCHROME 2
- RGB/YBR
- PALETTE COLOR

Note that if the color selected is not grayscale and the photometric interpretation being used is MONOCHROME 1 or MONOCHROME 2, then a grayscale representation for the selected pixel replacement color will be used.

See the [DICOM Anonymizer Conformance Statement](#) document included with the installed software for a complete breakdown of the supported image types for pixel data De-Identification.

4.7.5.4 One-Way Anonymization

For one-way anonymization of the patient demographic data and/or pixel data (re-identification cannot be performed), the following feature must be unchecked in the **Misc Settings** group box:

- **Enable Re-Identification**

If the **Enable Re-Identification** setting is enabled without a valid **Public Certificate**, the de-identification will fail with the following error:

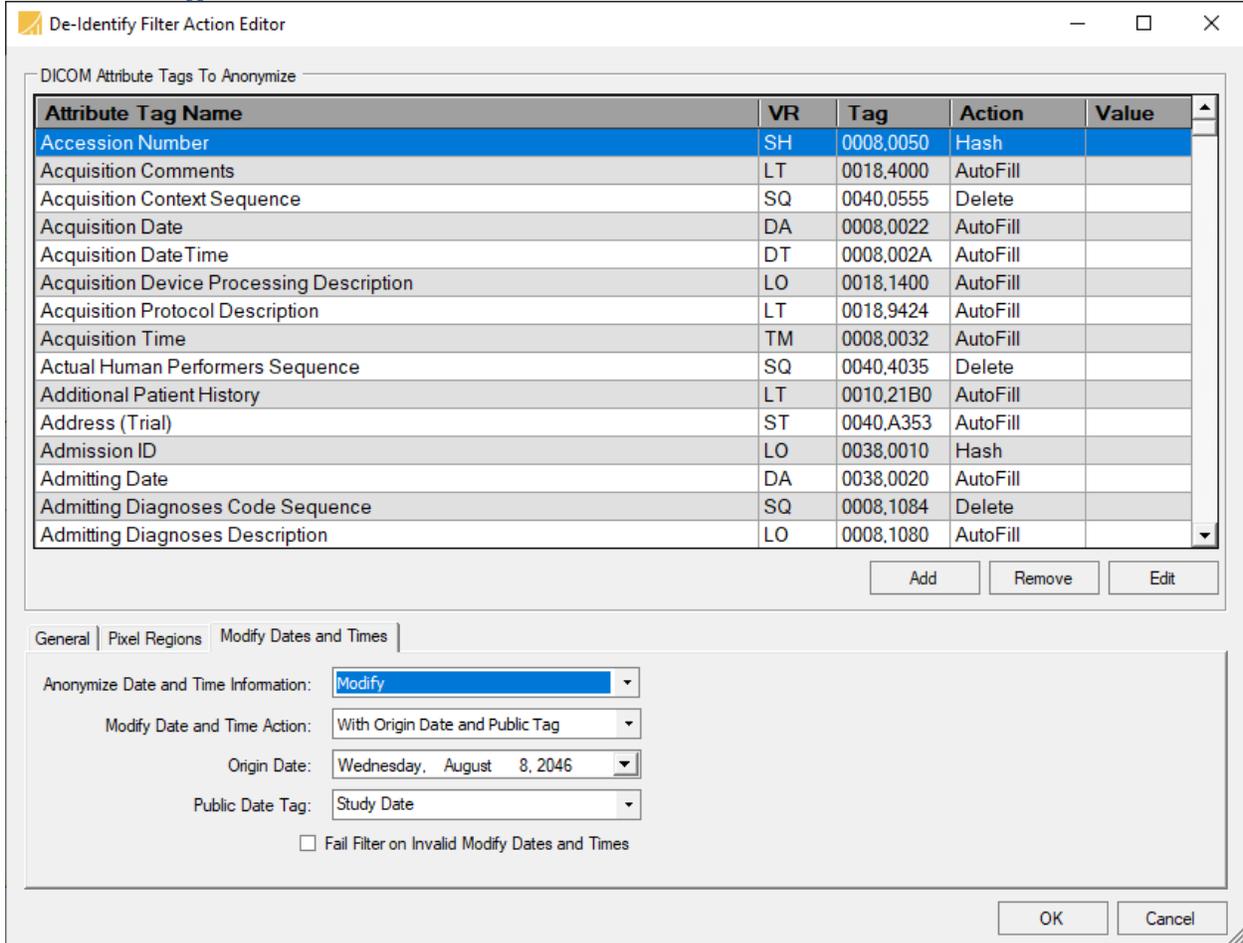
```
LaurelBridge.DCF.DCSEException: Failed to apply filter action --->  
System.Security.Cryptography.CryptographicException: Cannot find the requested object.
```

The **Public Certificate Path** text box will be disabled when performing one-way anonymization. A valid public certificate is only necessary when **Enable Re-Identification** is checked. A valid certificate should either be purchased or generated by the user.

The expected certificate file formats are:

- Public certificate (export format PEM): **<filename>.crt**
- Private certificate (export format PKCS #12): **<filename>.p12**

4.7.5.5 Modify Dates and Times



- **Anonymize Date and Time Information** – Retain Longitudinal Temporal Information Options:
 - **Retain** - All DA and TM elements in the data set to de-identify remain unchanged.
 - **Modify** - All DA and TM elements are modified in a manner that both retains the longitudinal temporal information, but still protects the PHI of the patient.
 - **Remove** - All DA and TM elements containing PHI are removed and replaced with applicable dummy values.
 - The legacy flag for 'Retain Date and Time Information' automatically maps to the 'Retain' option.

- **Modify Date and Time Action** – When Anonymize Date and Time Information is set to 'Modify', the following Modify Date and Time Actions are available:
 - **With Origin Date and Calendar Date** – Computes the number of days to shift date elements in a data set using two specified Dates: An Origin Date and a Calendar Date. This shift interval is then applied to each Date element in the data set to anonymize.
 - **With Origin Date and Public Tag** - Computes the number of days to shift date elements in a data set using a specified Origin Date and a date found at the indicated public Attribute Tag within the data set to anonymize. This shift interval is then applied to each Date element in the data set to anonymize.

NOTE: When using the 'Modify' option for Retaining Longitudinal Temporal Information, only the Date elements will be modified in the data set. This does include the date part of DT elements.

- **Origin Date** – The start date used to compute the number of days to shift date elements in a data set. This date can fall before or after the end date (given by the Public Date Tag), which can yield a negative shift interval. The example below will demonstrate this computation in further detail.
- **Public Date Tag** – The end date used to compute the number of days to shift date elements in a data set, given as a public Date tag in the data set to anonymize. The example below will demonstrate this computation in further detail.
- **Fail Filter on Invalid Modify Dates and Times** – If checked, the De-Identify filter will fail on invalid configuration data for the modify dates and times interval. When unchecked, the default modify dates and times interval will be used, which is a randomly generated timespan between 1970 and 2070.

Modify Date and Time Action with Origin Date and Public Tag Example:

Study Date: 2020/10/14

Content Date: 2020/10/20

Origin Date: 2046/08/08

Public Date Tag: 0008,0020 – Study Date

Formula: Public Date Tag Value – Origin Date = Computed Interval

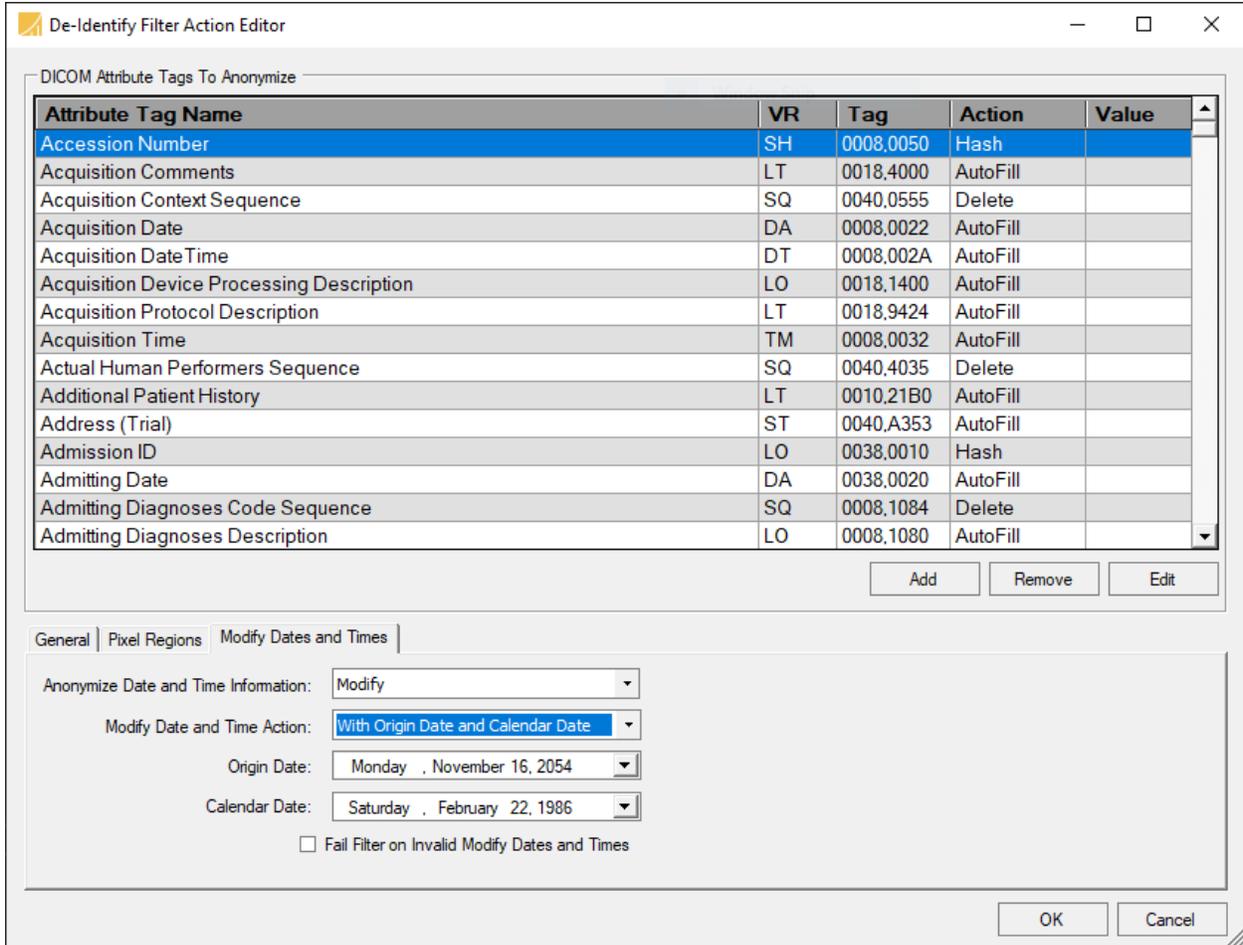
Computed Interval to modify dates and times: 2020/10/14 – 2046/08/08 = -16990 Days

Formula: Value to Modify + Computed Interval = Modified Date

Modified Study Date: 2020/10/14 + -16990 Days = 1974/04/09

Modified Content Date: 2020/10/20 + -16990 Days = 1974/04/15

Note that in the above example, each date in the data set to anonymize will be shifted by -16990 days, thus preserving the longitudinal temporal information between each date, while still anonymizing the actual PHI. You'll notice the days between the Study Date and Content Date remain 6 days, even after anonymization.



- **Origin Date** – The start date used to compute the number of days to shift date elements in a data set. This date can fall before or after the Calendar Date, which can yield a negative shift interval. The example below will demonstrate this computation in further detail.
- **Calendar Date** – The end date used to compute the number of days to shift date elements in a data set. This date can fall before or after the Origin Date, which can yield a negative shift interval. The example below will demonstrate this computation in further detail.

Modify Date and Time Action with Origin Date and Calendar Date Example:

Study Date: 2020/10/14

Content Date: 2020/10/20

Origin Date: 2054/11/16

Calendar Date: 1986/02/22

Formula: Calendar Date – Origin Date = Computed Interval

Computed Interval to modify dates and times: 1986/02/22 – 2054/11/16 = -25104 Days

Formula: Value to Modify + Computed Interval = Modified Date

Modified Study Date: 2020/10/14 + -25104 Days = 1952/01/21

Modified Content Date: 2020/10/20 + -25104 Days = 1952/01/27

Note that in the above example, each date in the data set to anonymize will be shifted by -25104 days, thus preserving the longitudinal temporal information between each date, while still anonymizing the actual PHI. You'll notice the days between the Study Date and Content Date remain 6 days, even after anonymization.

4.7.5.6 Additional Notes

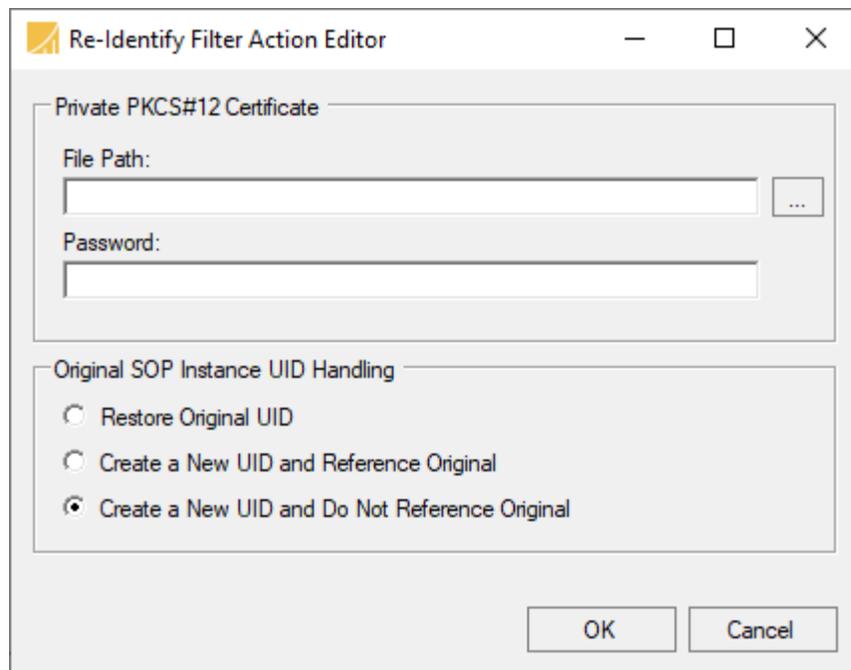
Pixel data and header data re-identification are only available with a valid public and private certificate.

Disable **Use Original Attributes Sequence** for all de-identification filters if you are going to anonymize SOP instances. Having this option enabled will defeat the purpose of anonymization by embedding the list of attribute tags and their original values that changed after applying the de-identification filter action.

Sequences are De-Identified per the following rules:

- If a given sequence (SQ element) contains an attribute tag that is marked for de-identification, the entire SQ element is anonymized. That is, each of the DICOM elements contained within that sequence are anonymized per either the specified tag replacement action, or AutoFill by default when determining the appropriate replacement value.
- If a sequence is marked as 'Delete', then the entire sequence is removed. A sequence will not be only partially altered. If any of the contents of a sequence are marked for de-identification, regardless of the action, then the entire sequence will be de-identified.

4.7.6 Re-Identify Filter Action



The **Re-Identify Filter Action** allows restoration of previously anonymized patient demographic information and pixel data for a given DICOM dataset.

The patient demographic information and pixel data restoration are only available as long as a valid private certificate (and password) is available. This private certificate must pair with the public certificate used to de-identify the original DICOM dataset.

Select one of the following actions when re-identifying a dataset to dictate how to handle the original SOP Instance UID:

- **Restore Original UID** – Select this option to restore the original SOP Instance UID. This option will not create a Referenced Image Sequence (0008, 1140).
- **Create a New UID and Reference Original** – Select this option in order to create a new SOP Instance UID and create a Referenced Image Sequence (0008, 1140) that references the original SOP Instance UID that was de-identified.
- **Create a New UID and Do Not Reference Original** – This option will create a new SOP Instance UID for the re-identified dataset without creating a Referenced Image Sequence (0008, 1140).

4.7.6.1 Additional Notes

Enable Logging – This option enables logging for the re-identification process.

Patient Identity Removed (0012, 0062) Attribute Tag will be added to the re-identified dataset in all cases, including selecting “Restore Original UID”.

As per **PS 3.15 Annex E Attribute Confidentiality Profile**, the following Attribute Tags are added when re-identifying a given DICOM dataset, unless restoring the original SOP Instance UID:

- Patient Identity Removed (0012, 0062)
- Purpose of Referenced Code Sequence (0040, A170)
- Referenced Image Sequence (0008, 1140)

4.7.7 Composer Action Examples

The Composer action uses .NET regular expressions to parse an element’s value and combine the parts into new elements.

- 1) **Swap two tags** – Swap the Patient’s Name and the Patient ID to be in each other’s place you would specify the following inputs:

Tag	Input Pattern
0010,0010	(.*)
0010,0020	(.*)

The regexes shown here mean that the entire value should be one capturing group. Then the outputs would look like this:

Tag	Output Pattern
0010,0010	\${2.1}

0010,0020	\${1.1}
-----------	---------

This results in the first pattern $\{1.1\}$ from the first input tag $\{1.1\}$ being put into the second output tag (0010,0020), and the first pattern $\{2.1\}$ from the second input tag $\{2.1\}$ being put into the first output tag (0010,0010). (In this case, the first pattern is also the entire value.) So, if you started with “John Doe” and “1.2.3.4.5” in Name and ID respectively, your result would be a Patient ID of “John Doe” and a Patient’s Name of “1.2.3.4.5”.

- 2) **Split one tag into two tags** –To take the Accession Number (0008,0050) and keep only the first 10 characters in it and put the rest of it into the Requested Procedure ID (0040,1001). In this case, the regex for the input pattern has to specify how to split the Accession Number.

Tag	Input Pattern
0008,0050	(^{10})(.*)

This regex means the first 10 characters will be the first capturing group and everything else will be the second capturing group. Then the outputs would look like this:

Tag	Output Pattern
0080,0050	\${1.1}
0040,1001	\${1.2}

This means that the first capturing group – the first 10 characters – will go into the Accession Number; everything else from the Accession Number will go into the Requested Procedure ID. If the initial Accession Number was “ABCDEF1234567890”, then you would have “ABCDEF1234” as the Accession Number and “567890” as the Requested Procedure ID. (Note that the output tag does not necessarily have to be parsed as an input.)

- 3) **Combine two tags** – To take parts of the Accession Number and parts of the Requested Procedure ID and “mix and match” them.

Tag	Input Pattern
0008,0050	(^{10})(.*)
0040,1001	(^{6})(.4)

These regular expressions mean to split the first tag into two capturing groups – the first one having 10 characters and the second one having whatever is left – and to split the second tag into two capturing groups, one of the first 6 characters and the second of the following 4 characters.

Then the outputs might look like this:

Tag	Output Pattern
0080,0050	\${1.1}---\${2.2}---\${2.1}
0040,1001	\${2.1}\${1.2}

If the initial Accession Number was “ABCDEF1234567890” and the initial Requested Procedure ID was “1.2.3.4.5.6.7.8.9.0”, then the resulting Accession Number would be “ABCDEF1234---4.5.---1.2.3.”; the resulting Requested Procedure ID would be “1.2.3.567890”. Note that the patterns can be used multiple times and also combined with plain text.

4.7.7.1 Working with DICOM sequences

A sequence may be entered as a tag by appending it to a numeric tag (the traditional group-element pair) with a period (“.”). You may also indicate an item in the sequence with “#” and the sequence item ID, followed by the tag indicating the sequence. There may be multiple sequences and sequence IDs as part of one “tag”. Examples are shown below:

- Simple tag - 0010,0010
- Tag within sequence - 0080,0100.0008,0060
- Tag within specific sequence item - 0080,0100.#0.0008,0060
- Tag within nested sequence with sequence items - 0080,0100.#1.0080,0100.#0.0008,0060

If no item number is specified, the first item (#0) is assumed. Specify the last element in a sequence by “#L” (upper-case is important!) if the number of sequence items is unknown. Specify the next item in the sequence via “#N” (again, case is important) to append to the sequence.

For example: 0080,0100.#L.0010,0010.#N.0008,0060

Please notice that:

- The sequence IDs (e.g., #1) and the tag-value pairs for the sequences are all separated by periods (“.”).
- The tags for the sequences are simple group-element pairs themselves.

4.7.7.2 Modifying private tags

Private DICOM elements can be modified using the **Composer Filter Action**. But first, you must add a private data dictionary to Compass so Compass can know the VR (Value Representation) and VM (Value Multiplicity) of the private tags.

In the C:\ProgramData\Laurel Bridge Software\Compass directory create a new directory called **dicom**. In that directory create a file called **ext_data_dictionary.txt**. Its full pathname will then be:

C:\ProgramData\Laurel Bridge Software\Compass\dicom\ext_data_dictionary.txt.

Note that in earlier versions of Compass, this file needed to be named just **ext_data_dictionary**, without the **.txt** extension. If this file name or path needs to be changed from the above defaults, please contact Laurel Bridge Software.

The contents of the file shall look like this:

```
#
# The following is an example extended data dictionary file.
#
[ elements ]
0029,1020 = CS,1,Example Private Attribute 1
0039,1020 = US,1,Another Example Private Attribute 2
0049,1001 = DS,1,Private DS attribute 3
0049,1002 = UL,1,Private UL attribute 4
0049,1003 = SL,1,Private SL attribute 5
0049,1004 = UI,1,Private UI attribute 6
```

A line beginning with a pound sign ‘#’ is a comment.

The line [elements] must be present and can exist only once in the file.

All other lines define private DICOM elements and should be specified in the format:

```
gggg,eeee = <VR>,<VM>,<Long text description>
```

where gggg is the group number and eeee is the element number,

followed by an equal sign,

followed by three comma-delimited fields: VR, VM, and the long text description.

For example, to add private tag 0029,1020 to the data dictionary with a VR of CS and a VM of 1, the ext_data_dictionary file should have the following line added under the [elements] line:

```
0029,1020 = CS,1,Example private attribute text description
```

You must restart the Compass Service after creating or modifying your private dictionary.

5 Routing of HL7 Messages

This section of the user manual will focus on how to configure Compass for routing of HL7 messages, including configuration of Sources, Destinations, Rules, and Job Actions. Note that many of these configuration areas have a

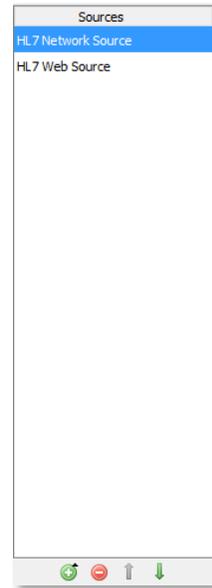


Description field; this field is a free-form text

box which can be used to hold user notes about the configuration item.

5.1 Creating an HL7 Source

A new Source can be created by selecting **Edit > HL7 Options...** from the menu and then selecting the **HL7 Sources** pane. Press the green plus button located under the Sources list. This will pop up a context menu with two choices: **HL7 Network Source** and **HL7 Web Source**. The list of Sources may be reordered by selecting a Source and then pressing the up or down arrows next to the green plus and red minus buttons.



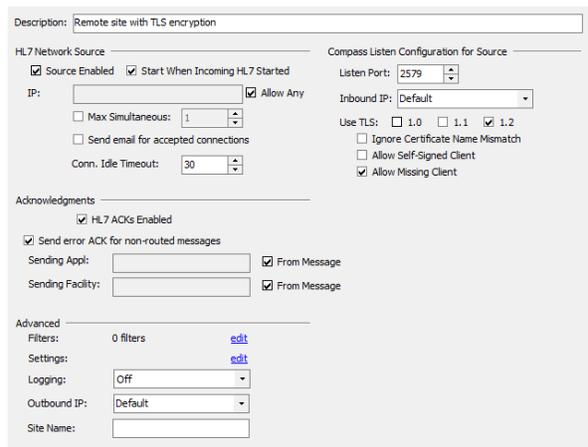
5.1.1 HL7 Network Source

An **HL7 Network Source** is any network device that will connect to Compass and send HL7 messages. Selecting **HL7 Network Source** will create a new Source with the name “New HL7 Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name. When a network sender connects with Compass, the Sources list is processed from top to bottom and the first match found is used (excluding HL7 Web Sources); therefore, the ordering of the Sources may be important if potentially more than one could match.

Once an **HL7 Network Source** has been created the next step is to configure its settings.

5.1.1.1 Settings

Under the **HL7 Network Source** heading, settings include configuration options for the remote **Host/IP**, the Enabled status of the Source, the autostart option, an option to limit the number of concurrent connections, and the type of responses that are sent back to the Source for non-routed messages. The **Allow Any** checkbox following the **Host/IP** can be selected to allow any value for that field. To limit the number of concurrent connections allowed by this Source check the **Max Simultaneous** checkbox and specify the desired number.



The enabled status for HL7 Sources works differently than for DICOM Destinations. For HL7 routing, it is desirable to be able to individually start and stop network traffic from a particular

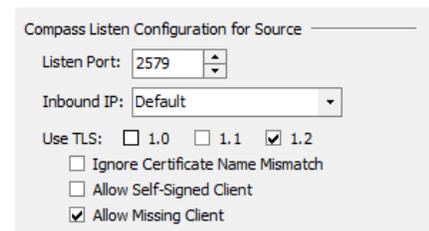
node. It may also be desirable to have a node configuration that exists but can never be started (e.g. during a maintenance cycle, or as an example configuration). As such, the following definitions apply:

- Enabled* – this means that the node is allowed to be started.
- Disabled* – this means that the node is not allowed to be started.
- Started* – this means that Compass has opened a listening socket on the specified IP/port for the source and either has an open network connection from that source or is waiting for such a connection.
- Stopped* – this means that Compass is not listening for connections from that source; any such connection attempt from the source will fail.

Given these definitions, the settings are easily understood. Selecting the **Source Enabled** checkbox enables the source (allowing it to be started) but does not actually start the source; deselect it to prevent the specified Source from starting. Select **Start When Incoming HL7 Started** to start this particular HL7 Source when the HL7 Incoming button on the main screen is started.

The type of response from Compass that will be sent to the Source can also be configured. HL7 acknowledgments can be enabled in the **HL7 ACKs Enabled** field. The acknowledgment code for non-routed messages can also be configured. Messages fail to get routed to a Destination if they don't match any of the defined Rules. Compass's default behavior is to report an error back to the Source for a non-routed image; this is indicated by selecting the checkbox **Send error ACK for non-routed messages**. By deselecting this checkbox, Compass will always report back to the Source that it received the message successfully. The Sending Application and Sending Facility for the ACK messages can also be controlled. The default behavior is to use the Sending/Receiving application/facility from a received message in the reply. The user can override this behavior by unchecking **From Message** and supplying a desired value.

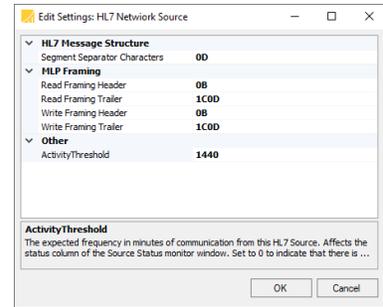
Under the **Compass Listen Configuration for Source** heading, a user specifies which network port Compass will use to listen for traffic from this source. In this way, HL7 Network Sources differ from DICOM Sources; each HL7 Network Source has its own unique listening port. The **Inbound IP** combo box is used to determine which system IP address Compass will use for listening purposes for this Source.



The **Use TLS** checkboxes specify whether network communications with the specified HL7 Source will be encrypted using TLS (as well as which TLS versions will be supported). The **Ignore Certificate Name Mismatch Errors** and **Allow Self-Signed Client Certificates** checkboxes can be used to relax the Compass certificate validation for this HL7 Source. However, we strongly recommend using these options for testing only, as they greatly reduce security by preventing full TLS authentication from occurring. The **Allow Missing Client Certificates** checkbox can be used to control whether or not client certificates (i.e., client authentication) are required (note that server authentication is always mandatory).

Using TLS also requires that the TLS certificate information be configured correctly in the **System** pane – see the **System** section for more information. See **Appendix C: Section 1.3 Configuring Secure HL7 Communication** for more details about using Compass TLS support.

Under the **Advanced** heading, a user can give more in-depth control over how a Source communicates with Compass. The **Filters** link is currently unavailable – it will be supported in future Compass releases. Data handling options and link-level framing characters are areas which can be configured in finer detail when editing the **Settings** options. Verbose logging can be turned on or off on a per Source basis by selecting ‘On’ or ‘Off’ in the **Logging** combo box. When sending an outbound job, the IP address Compass will use can be specified by the **Source** via the **Outbound IP** combo box, if and only if the Destination that the job is targeted for has its **Host/IP** combo box set to Default. The **Site Name** field is a free-form text field which is used to help the user group/view Sources and Destinations on the main user interface screen. The user may type any text in this field he desires. Later, when viewing the Sources, Destinations, or Active Connections in the HL7 Connections and Nodes panel, the user can search for text contained in the Source definitions in the **New Filter** area.



5.1.2 HL7 Web Source

An **HL7 Web Source** is a special source which applies when a user submits an HL7 message to Compass using the Web Interface’s **Order Entry** tab. This feature is described in more detail in the **System** and **Web Interface** sections. Essentially, to use the Web Order Entry feature, the user must create one or more HL7 Web Sources; later, the user configures message templates which apply for these HL7 Web Sources. Selecting **HL7 Web Source** will create a new Source with the name “New HL7 Web Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name.

Once an **HL7 Web Source** has been created the next step is to configure its settings. This configuration is essentially the same as for an HL7 Network Source, only much simpler. See the above section for more details.

5.1.3 HL7 Hot Folder Source

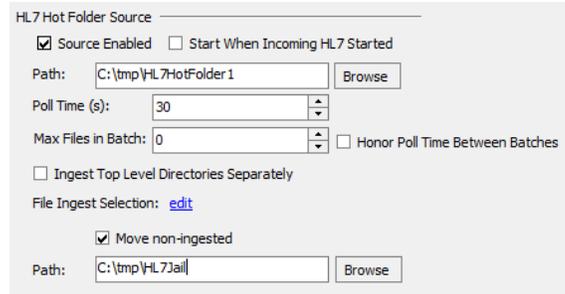
An “HL7 Hot Folder Source” is a Windows filesystem folder that Compass will continually monitor for text files each containing a single HL7 message, ingesting and filtering them, on a configurable polling interval. Selecting **HL7 Hot Folder Source** will create a new Source with the name “New HL7 Hotfolder Source-?”, where “?” is the next available number starting at 0. Source names are customizable and can be modified at any time by clicking on the name.

Note: it is the user’s responsibility to ensure that any and all data that may be transferred or transmitted using Compass is free of viruses, worms, Trojans, bots, or other code of a destructive or undesirable nature, collectively known as malware. Compass has no responsibility to scan or inspect in any way the data transferred or transmitted for the presence of malware and Laurel Bridge Software, Inc. has no liability of any kind associated with the transfer or transmission of such data.

Compass will delete the files from the hot folder after they have been processed by the HL7 Message Handler.

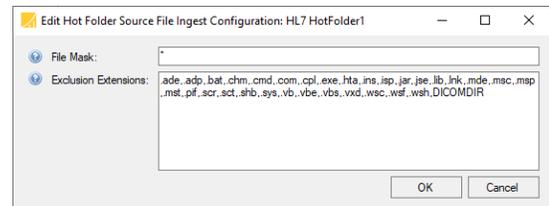
HL7 messages ingested via a hot folder match **HL7 Rules** in the same manner as HL7 Network Sources. Please note, the only exception is HL7 Hot Folder sources do not have an IP Address therefore you cannot match on the HL7 Connection IP Address.

Once an **HL7 Hot Folder Source** has been created the next step is to configure its settings.



5.1.3.1 Settings

Under the **Source** heading, settings include configuration options for the **Path** which should be monitored for HL7 messages, and **Poll Time**, which is the frequency to check the **Path** location for HL7 messages. There are batch control settings: **Max Files in Batch**, **Honor Poll Time Between Batches**, and **Ingest Top Level Directories Separately** (see [Appendix F: Hot Folder Basics / Definitions](#)). Additionally, the **File Ingestion Selection** restricts the hot folder to ingest or exclude specifically named files. The **File Ingestion Selection** link presents a dialog with two separate controls. The first, **File Mask**, is a regular expression that allows the user to select which files should be considered for ingestion. The second, **Exclusion Extensions**, provides a list of filename endings which will specifically be excluded. Exclusion takes precedence over inclusion. So, for example, if the user picks “*” for the **File Mask** (meaning “include everything” but uses the default exclusion list, no files with the default extensions (“.exe”, “.bat”, etc.) will be ingested.



There is also a checkbox that, if checked, moves non-ingested files out of the Hot Folder and into another user specified folder for later inspection.

The **Advanced** settings are the same as shown previously in section 5.1.1.1 for HL7 Network Sources.

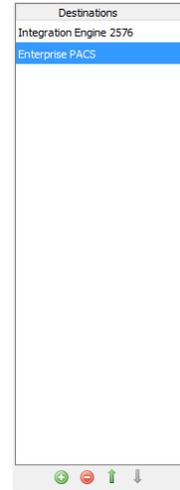
5.1.3.2 Zip File Ingestion

When scanning the contents of a hotfolder, if a valid zip file with an extension of “.zip” is encountered, Compass will extract the zip file and attempt to ingest the contents of the zip file as its own ingestion batch.

Please note that the decision to ingest the contents of a zip file is a function of the stability time of the zip file itself; the timestamps of the files contained in the zip file do not factor in to the stability time ingestion decision.

5.2 Creating an HL7 Destination

A new Destination can be created by selecting **Edit > HL7 Options...** from the menu and then selecting the **HL7 Destinations** pane. Press the green plus button located under the Destinations list. This will create a new Destination with the name “New HL7 Destination-?” where “?” is the next available number, starting at 0. Destination names are customizable and can be modified at any time by clicking on the name. The list of Destinations may be reordered by selecting a Destination and then pressing the up or down arrows directly below the Destinations list. Unlike Sources, there is no priority associated with Destination ordering; it is merely provided as a convenience.



Once an HL7 Destination has been created the next step is to configure its settings.

5.2.1 Compass Settings

Under the **Compass** heading, the only setting is the destination **Host/IP**.

5.2.2 Destination Settings

Under the **Destination** heading, settings include options for the **Host/IP**, the **Port** number, a **Test** button for testing network connectivity, the enabled status of the Destination, and a **Start When Outgoing HL7 Started** checkbox.

The enabled status for HL7 Destinations works differently than for DICOM Destinations. For HL7 routing, it is desirable to be able to individually start and stop network traffic to a particular node. It may also be desirable to have a node configuration that exists but can never be started (e.g. during a maintenance cycle, or as an example configuration). As such, the following definitions apply:

Enabled – this means that the node is allowed to be started.

Disabled – this means that the node is not allowed to be started.

Started – this means that the node either has an open network connection or Compass will create an open network connection as soon as there are messages ready to send to the node

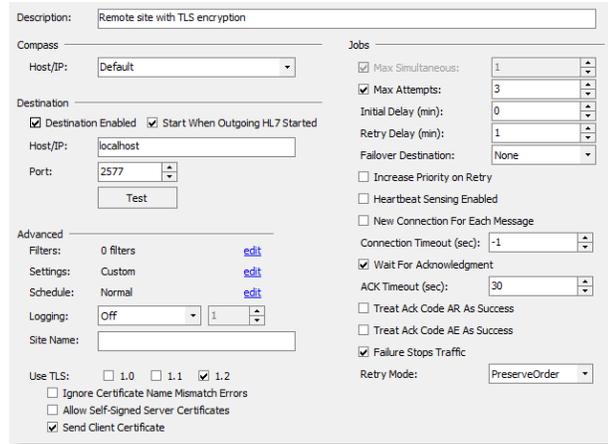
Stopped – this means that the node has no open network connection, and Compass will not create any network connection to this node until it is *Started*.

Given these definitions, the settings are easily understood. Selecting the **Destination Enabled** checkbox enables the destination (allowing it to be started) but does not actually start the destination; deselect it to prevent the specified Destination from starting. Select **Start When Outgoing HL7 Started** to start this particular HL7 Destination when the HL7 Outgoing button on the main screen is started.

Once values have been specified for the **Host/IP**, and the **Port**, then the **Test** button can be clicked to issue an ICMP Ping to the specified **Destination** machine. A green check next to the **Test** button indicates a successful Ping; a red X indicates a failure could not be made, or the connection was refused. Hover the mouse pointer over the **X** to read a tooltip with a message that further describes the failure.

5.2.3 Advanced Settings

Under the **Advanced** heading, a user can give more in-depth control over how **Compass** communicates with a **Destination**. **Filters** are not yet available and will be supported in a future **Compass** release. **Settings** options are configured in the same fashion as **Sources**. The **Schedule** specifies the time at which jobs are allowed to be sent to the **Destination**. A green box for the given day and time means jobs may be sent at that time; a red box means that jobs may not be sent at that time.



Each hour block represented by a green or red square may be toggled individually; alternatively, right-clicking on the schedule allows the schedule to be set to a commonly used setting. Be aware that choosing one of these predefined settings will replace the currently specified schedule. Verbose HL7 logging can be turned on or off on a per **Destination** basis by selecting **On**, **Off**, or **After Failures** in the **Logging** chooser. Selecting **After Failures** requires the number of failures before logging to be further specified. The **Site Name** options work the same way for **Destinations** as it does for **Sources** (see above).

The **Use TLS** checkboxes specify whether network communications with the specified **HL7 Destination** will be encrypted using **TLS** (as well as which **TLS** versions will be supported). The **Ignore Certificate Name Mismatch Errors** and **Allow Self-Signed Server Certificates** checkboxes can be used to relax the **Compass** certificate validation for this **HL7 Destination**. However, we strongly recommend using these options for testing only, as they greatly reduce security by preventing full **TLS** authentication from occurring. The **Send Client Certificate** checkbox can be used to control whether or not the client's certificate is sent to the server, enabling client authentication to occur as well (note that server authentication is always mandatory).

Using **TLS** also requires that the **TLS** certificate information be configured correctly in the **System** pane – see the **System** section for more information. See **Appendix C: Section 1.3 Configuring Secure HL7 Communication** for more details about using **Compass** **TLS** support.

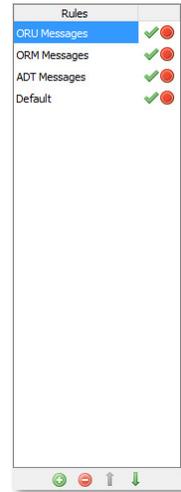
5.2.4 Jobs Settings

Under the **Jobs** heading, settings include options which affect how and when jobs are sent to the **Destination**.

The **Max Simultaneous** checkbox is not editable – it is displayed only to highlight a key difference between Destinations for DICOM and HL7 traffic. For HL7 destinations, there is only ever one network connection to a particular destination. The **Initial Delay** setting will delay jobs to this Destination by the specified number of minutes. The **Max Attempts** checkbox defines the maximum number of send attempts. If unchecked, the job will retry indefinitely. The **Retry Delay** will cause an unsuccessful job to be retried after the specified number of minutes. The **Failover Destination** setting allows a copy of the original job to be sent to the specified failover destination (if specified) once the original job has reached its **Max Attempts** without succeeding. Note that it is possible to configure an infinite loop of failover jobs, as Compass does not attempt to restrict the failover configuration. Checking the **Heartbeat Sensing Enabled** checkbox will cause Compass to verify that the Destination is available prior to attempting to send jobs to it. If the Destination is unavailable, Compass will queue up the jobs for the Destination without attempting to send them. If and when the Destination becomes available, Compass will begin sending the queued jobs. Compass issues an ICMP Ping message and inspects the reply (or lack thereof) to determine if the Destination is available. The **New Connection For Each Message** checkbox determines whether sending a message to this destination will reuse an already open network connection. Check this box to have Compass reconnect to the destination for each message. The **Wait For Acknowledgment** checkbox determines whether Compass will wait to receive an ACK response before sending the next message to the destination. Similarly, the **ACK Timeout** value specifies how long to wait (a value of 0 or -1 means “wait forever”). The **Treat Ack Code AR As Success** and **Treat Ack Code AE As Success** checkboxes allow Compass to flag the job state as Sent even if the response ACK contained AR or AE response codes (as opposed to the normal AA response code). The **Failure Stops Traffic** checkbox allows the user to prevent any additional messages to go to this destination if there are one or more failed messages in the Compass databases. If this box is checked, once one or more message reaches the specified **Max Attempts** and their job state is set to Failed, this destination cannot receive additional messages until all such failures have been corrected or removed. A destination which cannot receive messages because of this condition is flagged as “Blocked” on the Active Connections panel on the main screen. The **Retry Mode** combo box determines whether Compass should preserve order when a single attempt to send a message fails. Selecting **PreserveOrder** will cause Compass to retry the currently failed message until it succeeds or reaches its **Max Attempts**. Selecting **AllowReorder** will allow Compass to attempt other eligible messages for this destination when a failure occurs.

5.3 Creating an HL7 Rule

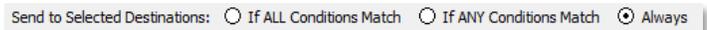
A new Rule can be created by selecting **Edit > HL7 Options...** from the menu and then selecting the **Rules** pane. Press the green plus button located under the Rules list. This will create a new Rule with the name “New Rule -?” where “?” is the next available number, starting at 0. Rule names are customizable and can be modified at any time by clicking on the name. The list of Rules may be reordered by selecting a Rule and then pressing the up or down arrows directly to the right of the Rules list. When Compass processes the list of Rules for each received image the Rules list is processed from top to bottom and the first match found is used, therefore the ordering of the Rules may be important if potentially more than one could match.



Once a Rule has been created the next step is to configure its Conditions, Selected Destinations, and Rule Options.

5.3.1 Conditions

A Rule’s Conditions determine whether or not its Actions will be



applied to an image being processed. A Rule may have multiple conditions, in which case it may be specified that either all the conditions must apply to the image or that only one condition must apply to the image in order for the Rule to match. Another option is that the Rule always matches; effectively declaring it to have no Conditions. Selecting one of three radio buttons, **If ALL Conditions Match**, **If ANY Conditions Match**, and **Always**, will implement one of these three scenarios.

The **HL7 Content** condition allows incoming images to be selected for this Rule based upon their contents.



When using this condition, the user specifies which piece of the HL7 message is being considered. The specified piece is described with a “query string” which takes one of the following forms:

```
XYZ | f  
XYZ | f . r  
XYZ | f . r . c  
XYZ | f . r . c . s
```

Where the pieces are:

- XYZ the three-letter code which indicates which segment in the HL7 message is being examined. Examples might include MSH, PID, ORC, OBR, etc.
- f a number, starting at 1, to indicate which *field* in the specified segment is being examined.
- r a number, starting at 1, to indicate which *repetition* in the specified field is being examined.
- c a number, starting at 1, to indicate which *component* in the specified repetition is being examined.
- s a number, starting at 1, to indicate which *subcomponent* in the specified field is being examined.

Example query strings:

- MSH|9 – the 9th field of the MSH segment
- PID|5.1 – the 1st repetition of the 5th field of the PID segment
- PID|5.1.2 – the 2nd component of the 1st repetition of the 5th field of the PID segment
- PID|5.1.2.3 – the 3rd subcomponent of the 2nd component of the 1st repetition of the 5th field of the PID segment

A typical encoding of an HL7 2.x message separates its contents into segments, one per line, starting with a 3-letter segment code. Usually, the pipe character (|) is used to separate fields within that segment. Each field can then be divided into repetitions, usually via the tilde character (~). Repetitions may be divided into components, typically via the caret character (^). Finally, components may be divided into subcomponents, usually with the ampersand character (&). One important note: in common HL7 parlance, components may be described via the segment, field number, and component number. For example: one might speak of “PID 5-2” to mean “Patient Name, First Name”. However, Compass needs to be able to allow users the ability to address particular repetitions of fields, even if the field has only one repetition in a particular message. As such, to address a particular component, the query string must specify the repetition number as well. So, in Compass, to address the 2nd component of a single-repetition Patient Name, the query string would be “PID|5.1.2”, even if only one repetition exists.

Consider the following example segment:

```
PID|1||597871^^^^EPI||ZZTEST^JOHN-
PUBLIC^Q||19611116|F|TEST^PATIENT^~ZZTEST^JOHNPUBLIC^F^|NH|1 SITENAME
```

The following table shows some examples of using query strings in the **HL7 Content** condition to access different parts of the message:

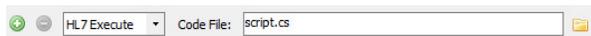
Query String	Value	Comment
PID 3	597871^^^^EPI	The entirety of field 3 in the PID segment

Query String	Value	Comment
PID 3.1	597871^^^^EPI	The entirety of the first repetition of field 3. Note that for this segment, this is the same value as PID 3, as there is only one repetition in this field (no ~ character).
PID 3.2	Error	This would be the second repetition of field 3, but since there is only one repetition present, this would be an error.
PID 3.1.1	597871	First component of first repetition of field 3
PID 3.1.5	EPI	Fifth component of first repetition of field 3
PID 4	<empty>	Field 4 is empty
PID 5	ZZTEST^JOHN-PUBLIC^Q	The entirety of field 5 in the PID segment
PID 5.1	ZZTEST^JOHN-PUBLIC^Q	The entirety of the first repetition of field 5. Note that for this segment, this is the same value as PID 5, as there is only one repetition in this field (no ~ character).
PID 5.1.1	ZZTEST	First component of first repetition of field 5
PID 5.1.2	JOHN-PUBLIC	Second component of first repetition of field 5
PID 5.1.3	Q	Third component of first repetition of field 5
PID 7	19611116	The entirety of field 7 in the PID segment
PID 8	F	The entirety of field 8 in the PID segment
PID 9	TEST^PATIENT^~ZZTEST^JOHNPUBLIC^F^	The entirety of field 9 in the PID segment
PID 9.1	TEST^PATIENT^	The entirety of the first repetition of field 9.
PID 9.2	ZZTEST^JOHNPUBLIC^F^	The entirety of the second repetition of field 9.
PID 9.2.2	JOHNPUBLIC	Second component of second repetition of field 9.

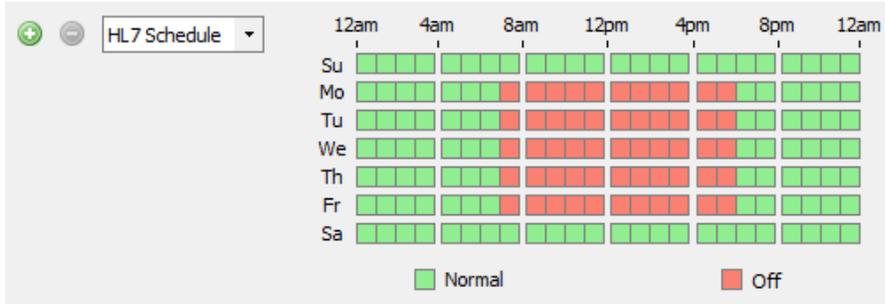
The **HL7 Connection** condition allows incoming messages to be selected for this Rule based upon their incoming HL7 Connection information.



The **HL7 Execute** condition allows incoming messages to be selected for this Rule based upon a custom, user-defined rule condition. See the example towards the end of this user manual for an example of a custom rule condition implementation.



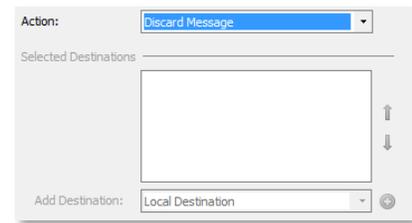
The **HL7 Schedule** condition allows incoming messages to be selected for this Rule based upon the time of day and day of week that the message was received. Any squares which are green in the schedule represent time/day blocks where the condition is true; any squares which are red represent time/day blocks where the condition is false. Each square can be changed individually by clicking on it. Alternately, the user can right-click to select from a menu of preset schedules for ease of populating the schedule as desired.



Configure the condition based upon the desired test. Conditions may be added or removed by pressing the green plus sign or the red minus sign located to the left of each condition.

5.3.2 Rule Actions

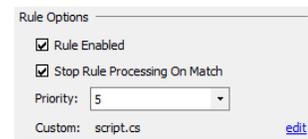
If all of the Conditions for the Rule pass, then the message currently being processed will have the currently selected action applied: **Discard Message** or **Send to Selected Destinations**. To add another Destination to the **Selected Destinations** list, select the desired Destination in the **Add Destination** combo box and press the green plus sign. To remove a Destination from the list, right-click on the desired Destination and choose **Remove Selected** in the context menu. A predefined, special Destination exists called the **HL7 Hold Queue**. Any jobs routed to the **HL7 Hold Queue** will wait there indefinitely until the user decides to perform an action on them, such as sending them to another Destination or deleting them.



5.3.3 Rule Options

Uncheck the **Rule Enabled** checkbox to disable the selected Rule. If unchecked, the Rule will be passed over during Rule processing.

Check the **Stop Rule Processing On Match** checkbox if no further Rules in the Rules list should be processed if this Rule matches (Rules are processed sequentially from the top of the list to the bottom of the list). If unchecked, Rule processing will proceed to the next Rule even if this Rule matches.



Specify a number in the **Priority** number chooser to assign the order in which the job that the specified image contains will be processed. 1 is the highest priority, and 10 is the lowest priority. If a job is populated with images that matched different rules containing different priorities, the job is assigned the highest priority of the matching rules. Jobs with a higher

priority will be sent before jobs with a lower priority. If a Destination has multiple jobs with the same priority, the older job is processed first.

In some cases, custom code extensions (such as Execute conditions) may want access to configuration data which is added to a particular rule. Click the “edit” link next to **Custom** to see the dialog where this configuration data is provided. In most cases, this data is not needed; when no configuration data has been provided, the rule displays the text “Not In Use”. Additionally, this Custom dialog provides a place to provide a code file which can run whenever this rule is successfully matched. If provided, the name of the code file is displayed here.

This **Custom** code file executes a piece of custom code that adheres to a particular interface:

```
using LaurelBridge.Compass.Core;
using LaurelBridge.HL7;

public class MyRuleMatchClass : IHL7RuleMatchAction
{
    void IHL7RuleMatchAction.Action(HL7RoutingRule ruleThatMatched,
        HL7Message receivedMessage, HL7Message responseMessage,
        HL7Acceptor acceptor, HL7JobRecord job)
    {
        // custom code
    }
}
```

5.4 Configuring an HL7 Job Action

Job Actions can be configured by selecting **Edit > HL7 Options...** from the menu and then selecting the **Job Actions** pane. Press the green plus button located under the Job Actions list. This will create a new Job Action with the name “New Job Action-?” where “?” is the next available number starting with 0. Job Action names are customizable and can be modified at any time by clicking on the name. The list of Job Actions can be reordered by selecting a Job Action and pressing the Up and Down arrows. The ordering of Job Actions in the list reflects their order of execution.

Once a Job Action has been created, the next step is to configure its Trigger, Type, Destinations, Code File, and Parameters.

5.4.1 Trigger

The Trigger can be set to one of the following options:

- **On Sent** – the action will be executed as jobs become marked as Sent
- **On Fail** – the action will be executed as jobs become marked as Failed
- **On Start** – the action will be executed each time a job enters the Running state

5.4.2 Destinations

A Job Action can be associated with jobs for all destinations, or a subset of configured destinations by using the checkboxes to make a selection.

5.4.3 Type

Execute

Allows custom actions to be defined

Code File

For Execute type actions the location of the file containing the code for the action must be specified. The button with the folder icon can be used to browse to the location of the code file.

Parameters

Execute actions support parameterization. Using the grid, a set of key/value pairs can be defined which will be passed into the action at runtime.

5.4.4 HL7 PowerScribe Custom Fields

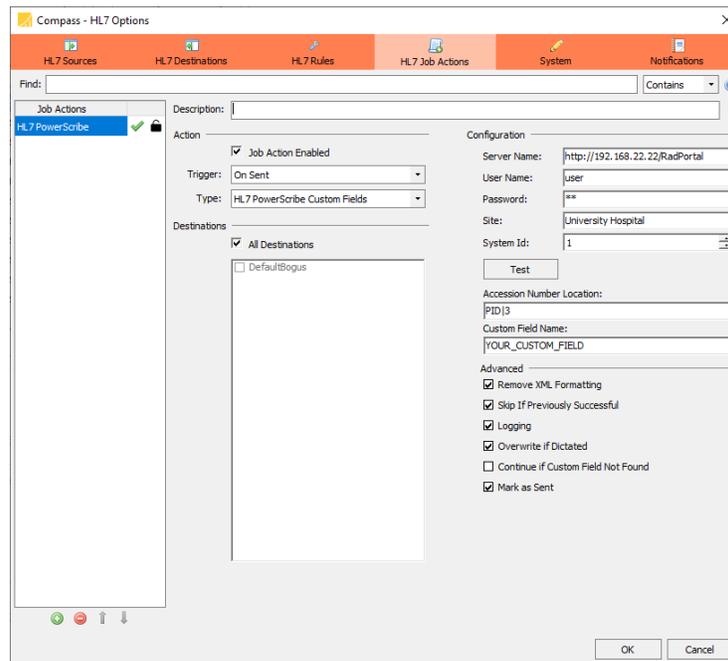
This job action requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM routing Compass. Note that this job action may be used for both PowerScribe 360 and PowerScribe One and that references below to PowerScribe 360 also apply to PowerScribe One.

The Compass HL7 PowerScribe Custom Fields Job Action extracts report data from an HL7 ORU message and maps it into the corresponding PS 360 Auto Text template custom field using

the PS 360's web API. A **Custom Field** name for the report data must be provided to insert the report text into a PS360 report.

All OBX segments will be scanned for report text. All text found in the 5th field will be concatenated together using the following rules:

- If explicit line breaks are found such as **\br** then they will be replaced with carriage return and new line characters in the text sent to PS360.
- If no explicit line breaks are found, then end-of-segment will be considered the correct location for a carriage return and new line characters.
- If XML formatting is contained in the report text it will be stripped according the **Remove XML** option from the job action's GUI.



Server Name

URL for the PowerScribe 360 Web API. For example,
`http://www.SomeURL.com/RadPortal`

User Name

Login name for the PS 360 Web API

Password

Password used to log on to the PS 360 Web API

Site

The site on the PS360 that has the reports that will be updated

Custom Field Name

Name of the custom field to be updated in PS360

Accession Number Location

Query string to locate the accession number in the current HL7 message. The accession number is typically in the 3rd field of the PID segment. The corresponding string used to extract this value is PID|3. See the section on HL7 Rule conditions for more query string examples.

Remove XML Formatting

If the report text can be parsed as XML then strip all XML formatting.

Skip If Previously Successful

If this job action has succeeded in the past do not rerun on a job retry.

Overwrite Dictated

If checked, attempt to overwrite auto text fields even if the report is in a state that indicates it may have already been dictated. The statuses that will be overwritten are Correction Rejected, Pending Correction, Corrected, Pending Signature, Sign Rejected, and Final. The PowerScribe system may prevent Compass from updating a report with one of these statuses. Compass will show a success status message even if the statuses could not be updated.

Continue if Custom Field Not Found

If checked, do not fail if the Custom Field cannot be found for the current Accession Number in the PS360 system. Either the HL7 message will be sent without error or the **Mark as Sent** option will apply.

Logging

Enable logging from job action

Mark as Sent

If the job action is successful mark the Compass job as sent without sending the HL7 message to its destination.

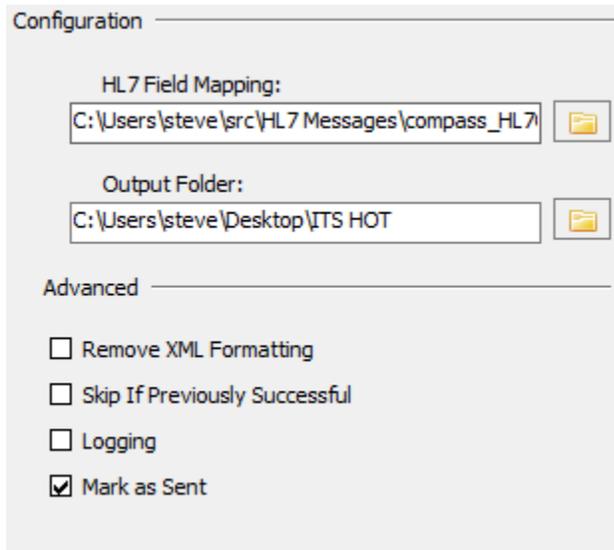
Sample Compass HL7 PowerScribe 360 workflow:

- 1) An HL7 message with a report is sent to Compass.
- 2) If the message is routed to a destination with the PS360 job action enabled the following steps will be followed:
 - a) Compass extracts the accession number from the HL7 message and uses it to query PowerScribe for all possible custom fields associated with that accession number.
 - i) If the accession number is found in PowerScribe go to step 2b
 - ii) If the accession number is not found in PowerScribe then the Compass job is marked as Failed and will be retried based on the current Compass configuration.
 - b) Compass will concatenate all report text from all OBX field 5 segments in the ORU message
 - c) Compass will loop through the Custom field names from PowerScribe for the given accession number
 - i) If the current PowerScribe Custom Field name has a mapping from the custom field name provided on the Compass job action GUI then Compass will set the value for the custom field in the first PowerScribe360 order found for that accession number.
 - ii) If there is no match, then do nothing.
 - d) The Compass job will (optionally) be marked as Sent without being forwarded to an HL7 receiver and will be purged by the Compass HL7 Sent job purger.

5.4.5 HL7 Report To DICOM SR

This job action requires a license. Contact Laurel Bridge Software to enable this feature on your HL7 routing Compass.

The Compass HL7 Report To DICOM SR Job Action extracts report data from an HL7 ORU message and creates a DICOM Basic Text SR. This is done by concatenating together all the OBX segments contained in the ORU. That SR will be written as a DICOM chapter 10 file to a directory on the file system. Typically, that directory will be a Compass HotFolder and the new DICOM object will be ingested and routed according to Compass DICOM routing configuration.



Configuration

HL7 Field Mapping:
C:\Users\steve\src\HL7 Messages\compass_HL7

Output Folder:
C:\Users\steve\Desktop\ITS HOT

Advanced

- Remove XML Formatting
- Skip If Previously Successful
- Logging
- Mark as Sent

HL7 Field Mapping

A text containing two columns of tab delimited text that maps HL7 fields to DICOM tags. The first column is the query string to use when searching the HL7 message. The second column contains a DICOM tag in the format of gggg,eeee where gggg is the hexadecimal representation of a DICOM group and eeee is the hexadecimal representation of a DICOM element.

Here are the contents of a sample file:

```
PID|5.1 0010,0010  
PID|3.1.1 0010,0020  
PID|7.1 0010,0030  
OBR|3.1.1 0008,0050  
ORC|9 0008,0020  
ORC|9 0008,0030
```

Output Folder

The full path to a folder where DICOM SR's will be written. The name of the file will be the SOP Instance UID of the DICOM SR object.

Remove XML Formatting

If the report text can be parsed as XML then strip all XML formatting.

Skip If Previously Successful

If this job action has succeeded in the past do not rerun on a job retry.

Logging

Enable logging from job action

Mark as Sent

If the job action is successful mark the Compass job as sent without sending the HL7 message to its destination.

6 System Settings

System settings can be configured by selecting either the **Edit > DICOM Options...** or **Edit > HL7 Options** from the menu and then selecting the **System** pane. Some options are universal across both the DICOM and HL7 facets of Compass; these options can be changed via either menu option. Options which apply to both DICOM and HL7 typically have a header which says “Compass” (e.g. “Compass Application Logging”).

6.1 DICOM System Settings

6.1.1 Compass Application Logging

Compass provides the ability to log a varying level of information to the application log files. The location, name, and size of the log files are configurable. The application log files are also implemented as rolling log files, meaning that when a log file reaches the specified maximum file size, a new application log file with a new name will be created and become the new application log. The rolling log file name has an incrementing number in the filename which increments each time the log file rolls over to the new file. It is also possible to configure how many of these rolling log files should be kept, as well as the verbosity of the log data. This is helpful for a variety of reasons, such as being able to see in detail what is being communicated between Compass and other devices.

Compass Application Logging / Unreferenced Session Log Purging

Directory: C:\ProgramData\Laurel Bridge Software\Col [Browse](#)

Filename: compass .X.log

Max File Size: 1024 KB

Number of Files: 10

Log Settings: Custom [edit](#)

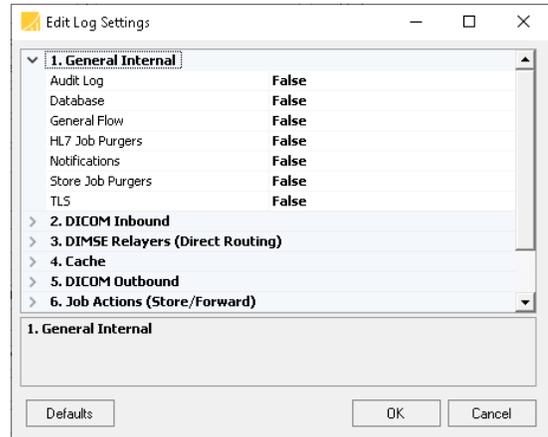
Session Logs Purge Delay 120 Minutes

The log files for the Compass Service use the root log file name as configured on the **Edit > DICOM Options > System** pane. The log files for the Compass Client use the root log file name with a “-client” suffix. The log files for the Compass Web Server use the root log file name with a “-web” suffix. The current log file for each Compass component can be viewed from the Compass Client by selecting **View > Current Application Logs** and selecting the desired log file.

The Session Logs Purge Delay specifies how many minutes to keep session log files before removing the logs.

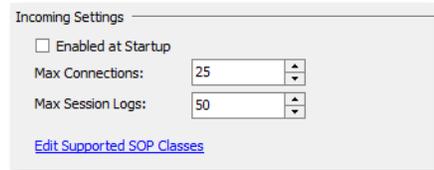
6.1.1.1 Log Settings

The [edit](#) link next to **Log Settings** selection allows the user to specify terse (False) or verbose (True) level for individual components within Compass. All entries are False when Defaults is selected. The Log Settings allow the user to fully control verbose logging for the specific areas of interest under investigation.



6.1.2 DICOM Incoming

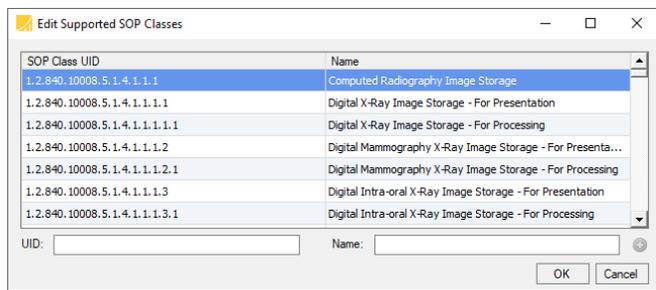
Compass will begin listening for incoming associations at launch using configured Listeners if the **Enabled at Startup** checkbox is selected.



Compass supports one or more Listeners for DICOM association requests as described in the [Creating a Listener](#) section. At least one Listener must be enabled.

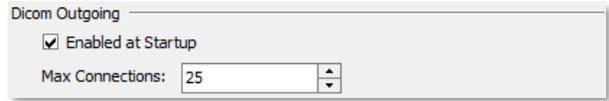
The maximum number of concurrent associations from Sources that Compass will process can be specified by using the **Max Connections** chooser. This is the absolute maximum number of incoming concurrent connections; if a Source has its individual maximum connections specified higher than this number then this number takes precedence.

The Store SOP classes accepted by Compass are configurable via the [Edit Supported SOP Classes](#) link. To add support for a new SOP class, enter its UID in the **UID** text field, its name in the **Name** text field, and press the green plus button. To remove support for a particular SOP class simply select the desired SOP class, right-click for the context menu, and choose the **Remove Selected** menu item. It is also possible to press the Delete key to remove the selected SOP class(es). To select a contiguous group of SOP classes for removal, select a SOP class by left-clicking it with the mouse and then select another SOP class while holding down the Shift key. To select a non-contiguous group of SOP classes for removal, select each individual SOP class while holding down the Ctrl key. Alternately, typing Ctrl+A will select all of the SOP classes in the table.



6.1.3 DICOM Outgoing

The maximum number of concurrent associations that Compass will use to connect to Destinations can be specified using the **Max Connections** chooser. This is the absolute



maximum number of outgoing concurrent connections; if a Destination has its individual maximum connections specified higher than this number then this number takes precedence.

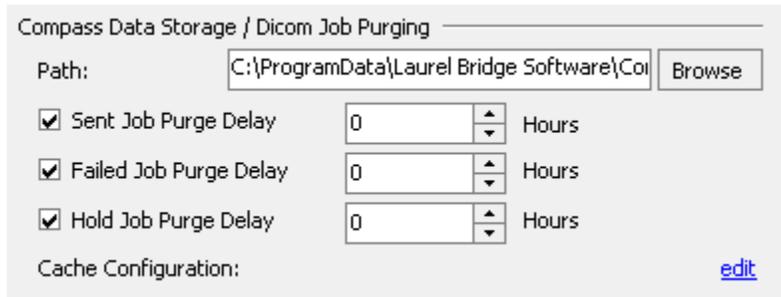
Compass will enable outgoing jobs at launch if the **Enabled at Startup** checkbox is selected.

6.1.4 Compass Data Storage

Compass provides the ability to specify the storage location for images. A different path may be specified from the default by typing in a new location or selecting it from the folder browser.

Note that the newly specified directory must already exist, and Input and Output must be stopped in order to change the location.

Compass will copy any existing images from the old storage location to the new location.

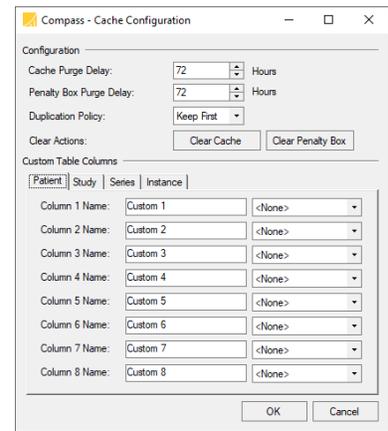


Compass is capable of purging itself of **Sent**, **Failed**, and **Hold** jobs after a configurable amount of time. Only **Sent** jobs older than the time threshold will be automatically purged by default.

The various **Job Purge Delay** controls specify how long to keep those types of jobs prior to removal.

6.1.4.1 Cache Configuration

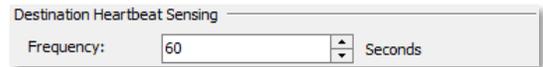
The **Cache** is a separate logical area within Compass where images are organized by the traditional Patient/Study/Series/Image hierarchy and can be queried via standard DICOM Query/Retrieve mechanisms. The **Cache** also contains within it a **Penalty Box** area for images which could not be added to the Cache (for example, because they contain a DICOM Study Instance UID that is already in the **Cache** but a Patient ID that is different than what is in the **Cache**). The Cache and the Penalty Box have their own **Cache Purge Delay** and **Penalty Box Purge Delay** which control how long images will be kept. When an image is to be added to the **Cache**, the **Cache** has to determine what to do with the header fields from that image if it is already present in the Cache. This behavior is determined by the **Duplication Policy**, which specifies whether to



keep the header fields from the first copy of the image or the most recently received image. The contents of the **Cache** and **Penalty Box** can be cleared using the Clear Cache and Clear Penalty Box buttons (these operations cannot be undone). Additionally, the **Cache** keeps header fields at each level of the Patient/Study/Series/Image hierarchy. In addition to obvious built-in fields such as Patient Name, Study Date, etc., the **Cache** can be configured with user-specified columns at each level using the **Custom Table Columns** control. Refer to the **Cache** section later in this manual for more information on how these configuration settings are used.

6.1.5 Destination Heartbeat Sensing

Each Destination defined in Compass has several ways of determining when jobs should be sent to that Destination. One way is to enable the

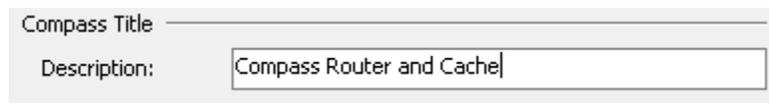


A screenshot of a configuration window titled "Destination Heartbeat Sensing". It contains a label "Frequency:" followed by a text input field containing the number "60" and a dropdown menu set to "Seconds".

Heartbeat Sensing feature on a given Destination. Each Destination that has this feature enabled will send a DICOM Verification request to the Destination at the frequency specified.

6.1.6 Compass Title

The Compass Title is displayed in the title bar of the Compass Client. The title is also displayed on the web browser tab when a connection is made to the Compass Web User Interface.

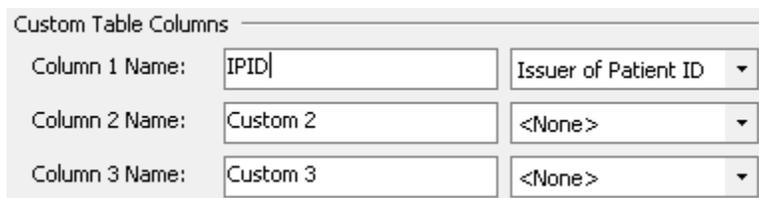


A screenshot of a configuration window titled "Compass Title". It contains a label "Description:" followed by a text input field containing the text "Compass Router and Cache".

The default value for Compass Title is the IP Address of the system hosting Compass.

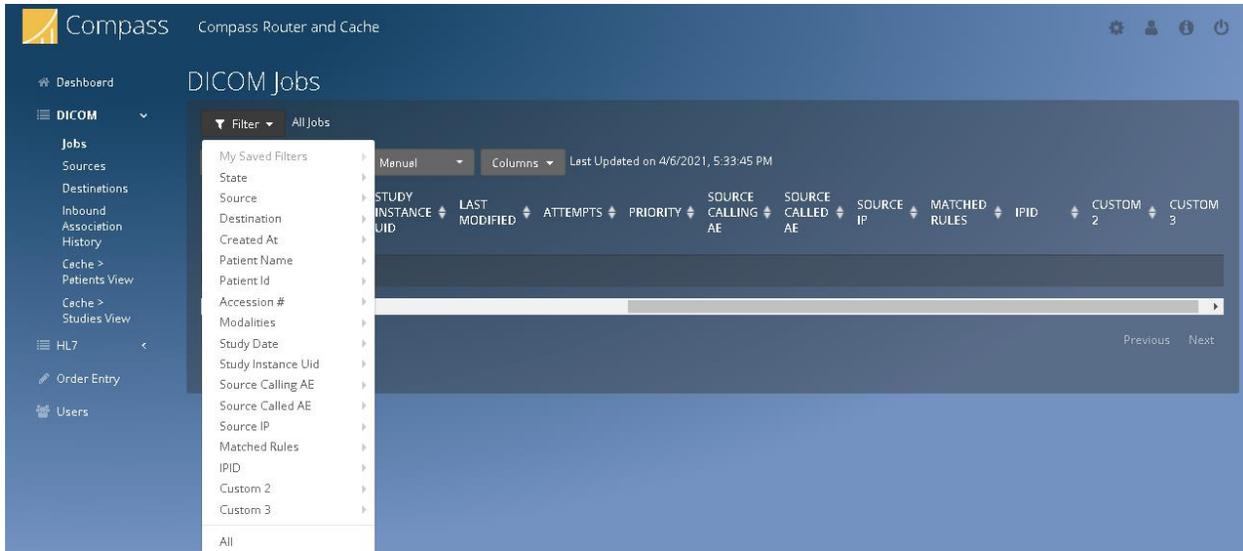
6.1.7 Custom Table Columns

The DICOM Jobs table on the Compass Web Interface is described in [section 11.3 DICOM/HL7 Jobs](#). In addition to the predefined columns, there are 3 Custom Columns that can be configured to match other values from the data stored for the job and displayed in the table. This example assigns



A screenshot of a configuration window titled "Custom Table Columns". It contains three rows, each with a label "Column X Name:" and two input fields. The first row has "IPID|" in the first field and "Issuer of Patient ID" in the second field with a dropdown arrow. The second row has "Custom 2" in the first field and "<None>" in the second field with a dropdown arrow. The third row has "Custom 3" in the first field and "<None>" in the second field with a dropdown arrow.

Issuer of Patient ID to Custom Column 1 with the name "IPID." Compass web user interface displays the filter and column as follows:



6.1.8 Compass Web Interface

Compass provides a web interface accessible via HTTP and/or HTTPS. You can customize the port numbers on which the interface will be hosted as well as the hostnames by which it is reachable. For example, if the computer that Compass is installed on is named `dicomrouter` and the HTTP web interface is configured for port 10400, then you would add `dicomrouter` to the **Hostnames** list. It would then be reachable with the URL `http://dicomrouter:10400`. If you wanted to specify the fully qualified name of the computer and your domain is `mycompany.com`, you would add `dicomrouter.mycompany.com` to the **Hostnames** list, and the URL would be `http://dicomrouter.mycompany.com:10400`. Note that the default port for HTTP (to avoid specifying a port number on the client side) is 80.

6.1.8.1 Secure Web Access (HTTPS)

The same is true for secure HTTPS connections to the Web service, except that the port number would be the specified port number in the “HTTPS:” setting, and the URL would begin with “https:”. Note that, in order to use HTTPS, you must specify the **TLS Certificate** and **Password** in the spaces provided (see **Appendix C: Secure DICOM and HL7 Communication with Compass** for more information). **Do not put the protocol portion of the address (i.e., “http://” or “https://”), in the Hostnames list.** Note that the default port for HTTPS (to avoid specifying a port number on the client side) is 443.

In order for the web server binding to exactly match the TLS certificate, the fully-qualified hostname must be entered into the **Hostnames** list exactly as given in the certificate. To see the hostname(s) given in the certificate, Control-click (hold Control while clicking on) the **TLS Certificate** label to bring up the X.509 certificate viewer, select the Details tab, then locate and click on one of the following fields:

- Click on “Subject Alternative Name” – The value(s) following each of the “DNS Name=” prefixes are the subject alternative names, which are the matching hostnames.

- If the above extension cannot be found, click on “Subject” – The value following the “CN=” prefix is the subject distinguished name, which is the matching hostname.

Note that it is possible for the certificate hostname to be wildcarded (in which case the certificate is what is known as a wildcard certificate). A wildcard certificate will match any hostname in the same subdomain as the wildcard value (“*”) given in the hostname. In this case, the actual fully-qualified hostname must be entered into the **Hostnames** list (i.e., not the wildcarded hostname).

If Compass is installed on Windows Server 2012, the **Disable HTTP/2** checkbox must also be checked if **HTTPS** will be used. The **Disable HTTP/2** checkbox can be left unchecked if only **HTTP** is used on Windows Server 2012.

6.1.9 Compass TLS Certificate Configuration

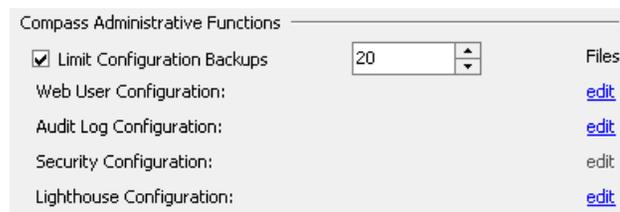
Using TLS requires that the TLS certificate information be configured correctly. The **TLS Certificate** should be set to the location of the certificate that Compass should present for identification to clients. This certificate will also be sent to servers if sending a client certificate has been enabled for any DICOM or HL7 destination. It is suggested that the certificate be a standard PKCS#12 certificate and it must contain an exportable private key. Finally, the **Password** must be set to the password for the private key in the certificate. Using a certificate format that does not password protect the private key allows this setting to be ignored (not recommended for security reasons).



See **Appendix C: Section 1.2 Configuring Secure DICOM Communication** for more details about using Compass TLS support.

6.1.10 Compass Administrative Functions

The following sections describe the configuration settings that are organized under **Compass Administrative Functions**.

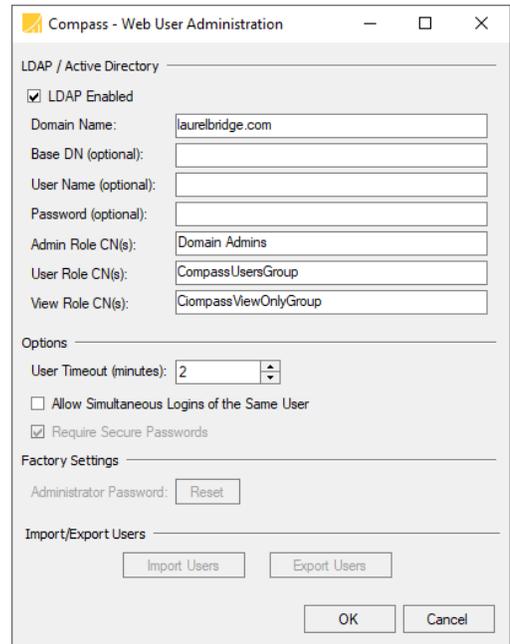


6.1.10.1 Limit Configuration Backups

When the Compass configuration is modified, Compass creates an automatic backup of the previous configuration (see **Appendix D:** for more information). This backup feature can be limited to a particular number of backup files by checking the **Limit Configuration Backups** checkbox and providing the maximum number of backup files to keep.

6.1.10.2 Web User Administration

For Web User Administration, select the **Web User Configuration edit** link. The **Compass – Web User Administration** dialog will be displayed as shown to the right. Compass supports **LDAP / Active Directory** for user account login to the Compass Web Interface. If enabled, the LDAP **Domain Name** can be specified. There are three optional fields: **Base DN**, **User Name** and **Password**. **Base DN** specifies the root from which all queries will be performed, i.e. “dc=example, dc=com”. **User Name** and **Password** are the credentials used to connect to the server. The three account types (as described below) will be mapped to the **User/Admin/View Role CN** (i.e., the LDAP Common Name, which, for Active Directory, corresponds to the Group Name) as configured. The CNs are comma-separated to allow for specifying multiple values that map to a single role.



If LDAP is not enabled, there are three types of built-in user accounts for the web interface: “Admin” accounts, “User” accounts and “ViewOnly” accounts. “Admin” and “User” accounts are allowed to affect existing jobs. Only “Admin” accounts can create and remove users.

The default “Admin” username for the web interface is “Administrator” and the password is “LaurelBridge1234”. It is highly recommended that you change the default password to a secure value (12+ characters, including uppercase and lowercase characters plus numeric digits) after successfully logging in. Since the “Admin” role allows modification of any user account, it is recommended that you only give access to your users as either “User” or “ViewOnly” accounts.

The default “ViewOnly” username for the web interface is “compass”, and the password is “Password1234”. It is highly recommended that you change the default password after successfully logging in. You should create “ViewOnly” users for anyone that is allowed to view compass jobs and any associated patient health information but not modify them.

There are no default “User” accounts. It is recommended that you create “User” level accounts for anyone authorized to affect changes on Compass jobs and view patient protected health information.

The **User Timeout** specifies the duration before a user logged in to the Web Interface will be logged out automatically. The **Allow Simultaneous Logins of the Same User** checkbox controls whether to enforce that users can only be logged on from a single browser (multiple tabs within the same browser count as a single logon). The **Require Secure Passwords** checkbox (enabled if LDAP is disabled) changes the minimum password length from the default 8 characters to 12 characters. It also adds the requirement that passwords contain at least one numeric digit (in addition to the default of both uppercase and lowercase letters). The **Administrator Password – Reset** button (enabled if LDAP is disabled) resets the password for the built-in “Administrator”

Admin user back to the default value given earlier. This password should be immediately changed to a non-default, secure value (12+ characters, including uppercase and lowercase characters plus numeric digits) after successfully logging in.

If not using LDAP and the sharing of local web users is desired, the **Import/Export Users** buttons can be used to transport user information between Compass instances. The **Export Users** button (enabled if LDAP is disabled) exports the list of built-in users (and their properties) to an XML file which can then be imported at a later time or on a different Compass system. The **Import Users** button (enabled if LDAP is disabled) imports these XML user lists. These buttons can also be used to backup and restore the local web users.

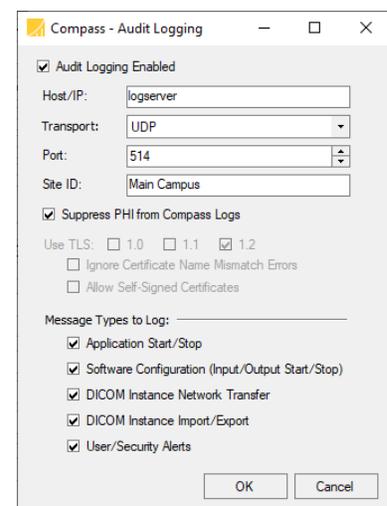
Because the user interface can display patient protected health information (PHI) when accessed, users must follow appropriate procedures to preserve the security of such information. It is highly recommended that the HTTPS interface be used (in favor of the HTTP interface). If the HTTP interface is in use, it is strongly recommended that it only be accessible from within your LAN or VPN. Furthermore, it is recommended that the **User Timeout** functionality discussed earlier be used to ensure that PHI does not stay visible on unattended screens (unless other similar security policies such as Windows auto-screen-lock policies are in place). Security of PHI is the responsibility of the organization using this software. Specific policies and practices to safeguard PHI are beyond the scope of this document.

6.1.10.3 Compass Audit Logging

To configure Compass audit logging capabilities, select the **Audit Log Configuration edit** link. The **Compass – Audit Logging** dialog will be displayed as shown to the right. This dialog allows the Compass audit logging capabilities (an implementation of DICOM PS 3.15 Appendix A.5 “Audit Trail Using Syslog” functionality) to be enabled or disabled. Compass audit log messages are typically sent to a remote (secured) syslog server, but copies of all audit log messages will also be stored locally in the Compass logs. This local audit log can be viewed from the Compass Client by selecting **View > Local Audit Log** and selecting the desired sort order.

If audit logging is enabled, the syslog server **Host/IP** address, **Transport** protocol, and **Port** can be configured. (Note that if the **Host/IP** address is left blank, audit logging will still be enabled locally.) The **Site ID** field allows configuration of the audit enterprise site ID, which is used to uniquely identify this Compass instance within the enterprise. Some of the copies of audit log messages that contain PHI may be suppressed from being stored in the Compass logs by enabling the **Suppress PHI from Compass Logs** checkbox.

If the **UDP** transport option is selected, the port defaults to the well-known port 514. If the **TCP** transport option is selected, the port defaults to the well-known port 601. If **TLS** transport is selected, the port defaults to the well-known port 6514. In addition, the supported version(s) of TLS can be configured, as well as whether or not to ignore certificate name mismatch errors and whether or not to allow self-signed certificates. Per RFC

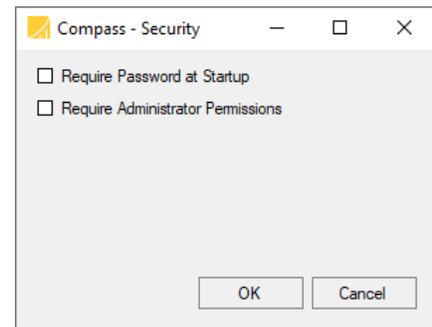


5424, use of the **TLS** transport option is strongly recommended unless the underlying network is secure.

Finally, the types of messages to be audit logged can be configured. Enabling **Application Start/Stop** messages will log a message each time the Compass application (both the Compass service and the client application) is started or stopped. Enabling **Software Configuration** messages will log a message each time an input or output (DICOM or HL7) is started or stopped. Enabling **DICOM Instance Network Transfer** messages will log a message each time a DICOM study network transfer is started or completed. Enabling **DICOM Instance Import/Export** messages will log a message each time a DICOM study is imported (from a hot folder) or exported (to a disk folder). (Note that it is possible to get multiple messages per DICOM study when importing, due to the inherent ambiguity of when a study is complete, as defined by the hot folder stability timeout.) Enabling **User/Security Alerts** will log a message whenever a user logs in/out, whenever a user is added/removed/modified, and whenever a DICOM/HL7 job is cloned/modified/removed.

6.1.10.4 Compass Security

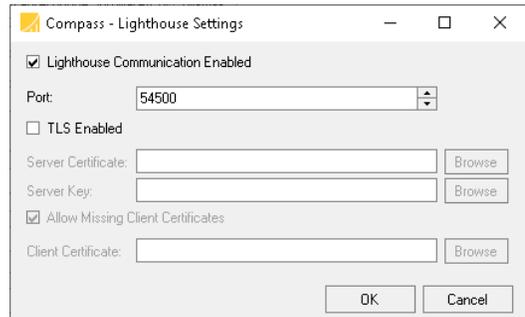
To configure Compass security capabilities, select the **Security Configuration edit** link. The **Compass – Security** dialog will be displayed as shown to the right. This dialog allows the Compass security capabilities to be enabled or disabled. If **Require Password at Startup** is enabled, the Compass Client will require entry of the valid password for the currently-logged-in Windows user before starting up. This prevents unauthorized users from being able to access the Compass Client if the console on the host Windows system is left unlocked. This setting is recommended if the Windows console is not kept locked whenever not in use.



If **Require Administrator Permissions** is enabled, the Compass Client will require administrator permissions before starting up (i.e., it must be started by right-clicking the executable or shortcut and selecting “Run as administrator”). Note that this has the additional, beneficial side effect of locking down the Compass data directory (as entered on the **System** pane, section **Compass Data Storage**, as the **Path** – the default value is “C:\ProgramData\Laurel Bridge Software\Compass”), so that only users with administrator privileges can view or modify Compass data, including temporary DICOM images (which may contain PHI) and Compass configuration and log files. This setting is recommended if any users of the host Windows system are not authorized to view PHI.

6.1.10.5 Lighthouse Configuration

To configure Compass communication with Lighthouse, select the [Lighthouse Configuration edit](#) link. The **Compass – Lighthouse Settings** dialog will be displayed as shown to the right. This dialog allows the Compass Lighthouse Communication to be enabled or disabled.



If Lighthouse communication is enabled, the **Port** can be configured. The default port is 54500.

If **TLS** transport is selected, the Server Certificate and Server Key become enabled. Click on the corresponding Browse button to open the file chooser dialog to select the Server Certificate and Server Key files. Finally, an optional Client Certificate can be selected when Allow Missing Client Certificates is disabled.

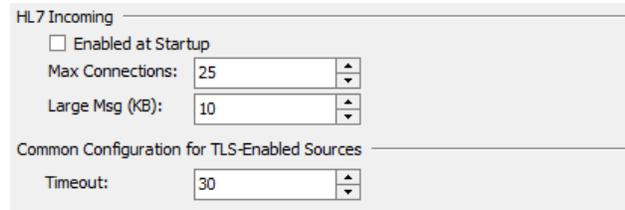
6.2 HL7 System Settings

6.2.1 Compass Application Logging

See above section under DICOM System Settings.

6.2.2 HL7 Incoming

Compass will begin listening for incoming HL7 connections at launch if the **Enabled at Startup** checkbox is selected. Only sources which are configured to **Start When Incoming HL7 Started** will be able to receive connections as a result of this setting. Any sources which are not so configured must be manually started to receive connections.



The maximum number of concurrent connections from Sources that Compass will process can be specified by using the **Max Connections** chooser. This is the absolute maximum number of incoming concurrent connections; if a Source has its individual maximum connections specified higher than this number, then this number takes precedence.



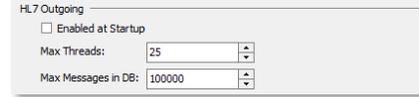
Compass typically stores HL7 messages in its database, but for performance reasons, certain messages which are sufficiently large will be stored in the file system. The large message threshold size can be set using the **Large Msg (KB)** selector.

The **Common Configuration for TLS Enabled Sources** area may be used to configure the TLS timeout settings for any HL7 Source which is configured to **Use TLS**, allowing incoming encrypted HL7 connections to Compass. Typically, such connections are from another copy of the Compass application. Also, to properly configure incoming TLS connections, the **TLS**

Certificate and/or **Password** must be set under the **Compass TLS Certificate Configuration** section. See the corresponding section under DICOM System Settings.

6.2.3 HL7 Outgoing

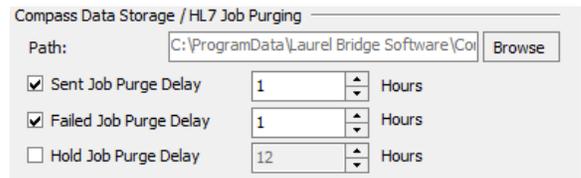
The maximum number of concurrent connections that Compass will use to connect to Destinations can be specified using the **Max Threads** chooser. This is the absolute maximum number of outgoing concurrent connections; if a Destination has its individual maximum connections specified higher than this number then this number takes precedence. The maximum number of HL7 messages which Compass will hold in its database can be configured using the **Max Messages in DB** selector. Once Compass has these many messages, the Compass HL7 Incoming will be turned off. Thereafter, if messages are removed or purged back to 90% of the maximum, the Compass HL7 Incoming will be turned back on.



Compass will enable outgoing jobs at launch if the **Enabled at Startup** checkbox is selected. Only destinations which are configured to **Start When Outgoing HL7 Started** will be able to start connections as a result of this setting. Any destinations which are not so configured must be manually started to receive connections from Compass.

6.2.4 Compass Data Storage

Compass provides the ability to specify the storage location for large HL7 messages which are written to file. A different path may be specified from the default by typing in a new location or selecting it from the folder browser. Note that the newly specified directory must already exist, and Input and Output must be stopped in order to change the location. Compass will copy any existing large HL7 messages from the old storage location to the new location.



Compass is capable of purging itself of **Sent**, **Failed**, and **Hold** jobs after a configurable amount of time. Only **Sent** jobs older than the time threshold will be automatically purged by default. The various **Job Purge Delay** controls specify how long to keep those types of jobs prior to removal.

6.2.5 HL7 Destination Heartbeat Sensing

Each Destination defined in Compass has several ways of determining when jobs should be sent to that Destination. One way is to enable the Heartbeat Sensing feature on a given Destination. Each Destination that has this feature enabled will send a DICOM Verification request to the Destination at the frequency specified.



6.2.6 Compass Title

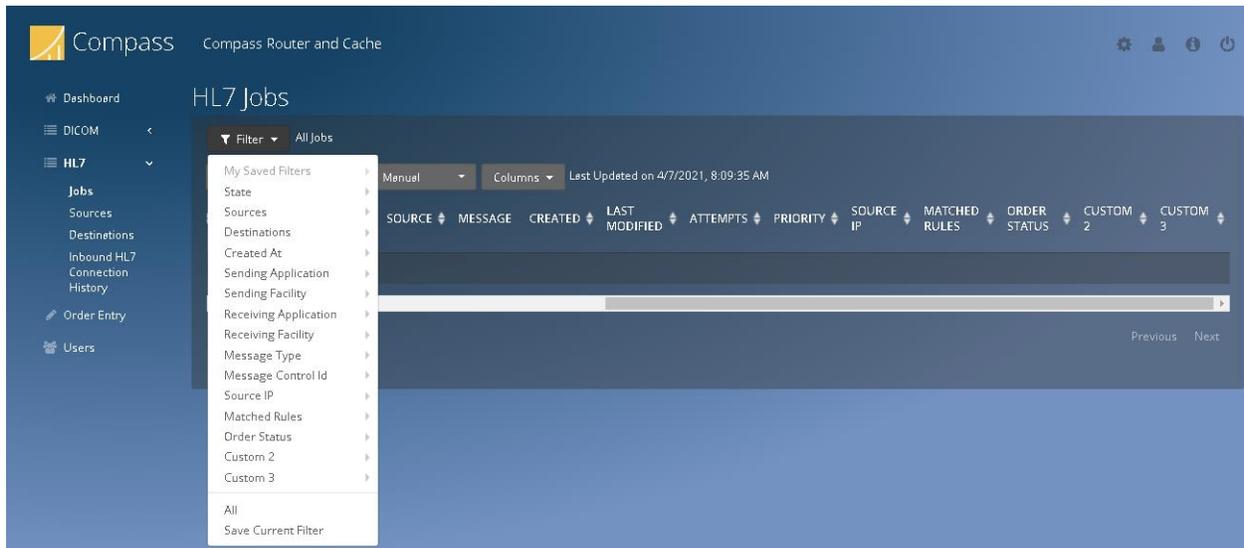
See above section under DICOM System Settings for information about how to configure the Compass title.

6.2.7 Custom Table Columns

The HL7 Jobs table on the Compass Web Interface is described in [section 11.3 DICOM/HL7 Jobs](#).

In addition to the predefined columns, there are 3 Custom Columns that can be configured to match other values from the data stored for the job and displayed in the table. This example assigns ORC | % to Custom Column 1 with the name “Order Status.” Compass web user interface displays the filter and column as follows:

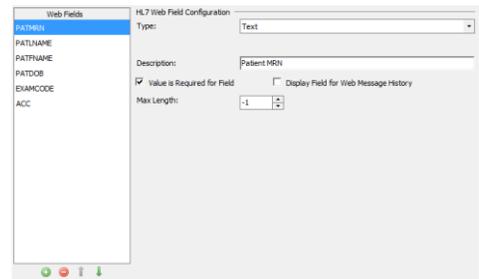
HL7 Custom Table Columns		
Column 1 Name:	<input type="text" value="Order Status"/>	<input type="text" value="ORC 5"/>
Column 2 Name:	<input type="text" value="Custom 2"/>	<input type="text"/>
Column 3 Name:	<input type="text" value="Custom 3"/>	<input type="text"/>



6.2.8 Compass Web Interface / HL7 Messaging via Web

See above section under DICOM System Settings for information about how to configure the Compass web interface.

This section of the HL7 System Settings handles the configuration options for the [HL7 Messaging via Web](#) feature. This feature allows the user to submit HL7 Messages to Compass through the Compass Web Interface. The user must define the format of the messages to submit using [Fields](#) and [Templates](#).



A **Field** is simply a portion of a message. **Fields** can have one of four types:

- Field type: **Text** – a simple text field, with an optional Maximum Length. For example, a user might create a field name “PATMRN” which is a Text field containing a patient medical record number.
- Field type: **Date** – a date field with year, month, and day. For example, a user might create a field named “PATDOB” which is a Date field containing the patient date of birth.
- Field type: **List** – a field with a set of known values which come from a file. For example, a user might create a field named “EXAMCODE” which is a List field with a value from a file of known exam code values. Each line in the file must have either a single value with nothing else, or a value, then a tab character, and then an optional description.

Example file contents:

```
Value1  
Value2  
Value3
```

Or

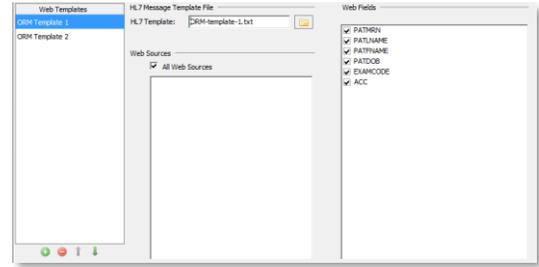
```
Value1  This is Value 1  
Value2  This is Value 2  
Value3  This is Value 3
```

[NOTE: there is an embedded tab character in each row between ValueX and “This”].

- Field type: **Execute** – a field whose value is provided by running code which implements a known code interface. A minimal example of this is:

```
using System.Collections.Generic;  
using LaurelBridge.Compass.Core;  
  
public class MyExecuteClass : IHL7WebFieldValueProvider  
{  
    public string GenerateValue(HL7WebTemplateDefinition webTemplateDef,  
        HL7WebFieldDefinition webFieldDef,  
        Dictionary<string, string> mappingDict)  
    {  
        // Custom code to return value here  
        return "some generated value";  
    }  
}
```

Fields are created using the familiar green plus icon. Each field has a check box to indicate whether the field is “required”, which means the user must enter a value for the field when submitting the message via the web. There is also a check box for **Display Field for Web Message History**. When checked, the value of this field will appear in the web UI as part of the description for previously submitted messages. See the Web UI section below for more details.



Once **Fields** have been created, the user can create **Templates**. A Template is simply a skeleton of an HL7 message which needs to have some values added to it. Templates must specify the text file containing the body of the message being filled out, which Web Fields they utilize, and also which Web Sources may submit messages of the particular template type.

Here is a sample template file:

```
MSH|^~\&|Compass|MyHospital|${WEBTEMPLATENAME}|${WEBSOURCENAME}|20150
101151633||ORM^O01|Q1881519657T13762602|T|2.3

PID|1||${PATMRN}^^^^^ABC||C230^PACS TRAIN||${PATDOB}|F||1600 PENNA
AVE^^WASHINGTON^DC^20500^US^home^^MyCounty|(800) 555-
4444^Home||||320034535131

OBR|1|${ACC}^Order Number|700562392^PACS ID|${EXAMCODE}^DX Whatever

OBX|1|TS|Requested Start Date/Time^Requested Start
Date/Time||20150511151600
```

The template file can be filled in with any value of a known Field using the syntax `${FIELDNAME}`. In the above example, the fields “PATMRN”, “PATDOB”, “ACC”, and “EXAMCODE” are used. In addition, there are several other built-in functions which can be used:

- `${WEBTEMPLATENAME}` is the configured name of the Template in use
- `${WEBSOURCENAME}` is the configured name of the Web Source submitting the message
- `${TIMESTAMP (yyyymmddHHmmss)}` - A timestamp that can be inserted into the message
- `${GUID ()}` is a unique ID that can be inserted into the message

6.2.9 Compass TLS Certificate Configuration

See above section under DICOM System Settings for information about how to configure the Compass TLS certificate.

6.2.10 Compass Administrative Functions

See above section under DICOM System Settings for information about how to configure Compass administrative functions.

6.2.10.1 Limit Configuration Backups

See above section under DICOM System Settings for information about how to limit configuration backups.

6.2.10.2 Web User Administration

See above section under DICOM System Settings for information about how to configure Compass Web Users.

6.2.10.3 Compass Audit Logging

See above section under DICOM System Settings for information about how to configure Compass audit logging.

6.2.10.4 Compass Security

See above section under DICOM System Settings for information about how to configure Compass security.

6.2.10.5 Lighthouse Configuration

See above section under DICOM System Settings for information about how to configure Compass Lighthouse settings.

7 Notifications

Notifications can be configured by selecting either the **Edit > DICOM Options...** or **Edit > HL7 Options** from the menu and then selecting the **Notifications** pane.

Some options are universal across both the DICOM and HL7 facets of Compass **Notifications**; these options can be changed via either menu option. Options which apply to both DICOM and HL7 typically have a header which says “Compass” (e.g. “Compass Email Properties”).

DICOM and HL7 Notifications can be sent for job failures or long running jobs (jobs that take too long to send). Additionally, DICOM notifications can be sent for low disk space.

If the **Failed Jobs Frequency** checkbox is checked, Compass will send an email detailing the failed jobs to the Notification Recipients at the specified frequency. The job information contained in the emails can be configured with the **Email Columns** chooser.

If the **Long Running Jobs Threshold** checkbox is checked, Compass will send an email detailing the long running jobs to the notification recipients. Only jobs that have been in the **Running** state longer than the specified threshold fall into this status.

If the **Queued Jobs Threshold** checkbox is checked, Compass will send an email detailing the **Queued** jobs count that exceeded the threshold. The job information contained in the emails can be configured with the **Email Columns** chooser.

If the **Heartbeat fails and reconnects** checkbox is checked, Compass will send an email detailing the heartbeat failure or reconnection to the notification recipients. This notification is only sent for **Destinations** with **Heartbeat Sensing** enabled.

The screenshot shows the 'Job Notifications' configuration window. It is divided into several sections. The top section, 'Job Notifications', contains four checkboxes: 'Failed Jobs Frequency (minutes)' (checked, value 15), 'Long Running Jobs Threshold (minutes)' (checked, value 60), 'Queued Jobs Threshold' (checked, value 1000), and 'Heartbeat fails and reconnects' (unchecked). Below this is the 'Email Columns' section, which is a list box containing: 'Last Modified' (checked), 'Patient Name' (checked), 'Patient Id' (checked), 'Destination' (checked), 'Message' (checked), 'Study Uid' (checked), 'Accession Number' (checked), 'Study Date' (checked), and 'Modalities' (checked). The next section is 'Low Disk Space Triggers', which has two sub-sections. The first sub-section is 'Notifications Enabled' (checked) with 'Frequency (minutes):' set to 60 and 'Threshold (%)' set to 20. The second sub-section is 'Input Cutoff Enabled' (checked) with 'Frequency (minutes):' set to 32 and 'Threshold (%)' set to 12. The final section is 'Low Cache Space Triggers', which has 'Notifications Enabled' (checked) with 'Frequency (minutes):' set to 60 and 'Threshold (%)' set to 10.

If the **Low Disk Space Notifications Enabled** checkbox is checked and the computer has less disk space available than the notification threshold, Compass will send an email to the Notification Recipients at the specified notification frequency.

If the **Low Disk Space Input Cutoff Enabled** checkbox is checked and the computer has less disk space available than the input cutoff threshold, Compass will automatically disable the DICOM inputs. The available disk space checks will be made at the specified input cutoff frequency. If the user attempts to restart the DICOM inputs while the available disk space is still below the input cutoff threshold, the user will be warned prior to reenabling the inputs.

If the **Low Cache Space Notifications Enabled** checkbox is checked and the system has less available number of cache images than the notification threshold, Compass will send an email to the Notification Recipients at the specified notification frequency.

To add a new recipient, press the green add button located beneath the list of **Notification Recipients**. Specify the recipient's email address and press the Enter key. The sender of the email can be specified as well.

It is necessary to include a valid **Mail Server Configuration** in order to send emails to the listed recipients. At a minimum, a **Host** and **Port** are required, and optionally TLS can be enabled by selecting the **Enable TLS** checkbox. A **Username** and **Password** are also required if the mail server requires authentication. Emails may also optionally contain a provided **Subject Prefix** and/or **Subject Suffix**.

The image shows two overlapping dialog boxes from the Compass software. The top dialog, titled "Compass Email Properties", has three input fields: "From:" with the value "compassadmin@yourdomain.net", "Subject Prefix:" which is empty, and "Subject Suffix:" which is empty. Below it is a section titled "Compass Notification Recipients" which contains a large empty rectangular area. At the bottom of this section are two small circular buttons, one green with a plus sign and one grey with a minus sign. The bottom dialog, titled "Compass Mail Server Settings", has several fields: "SMTP Server:" with the value "localhost", "SMTP Port:" with a dropdown menu showing "25", an "Enable TLS" checkbox which is unchecked, "Auth Mode:" with a dropdown menu showing "SMTP", "Username:" which is empty, and "Password:" which is empty. A "Test" button is located to the right of the "Auth Mode:" dropdown.

8 Enabling Input and Output

8.1 DICOM Input and Output

To allow Compass to accept incoming DICOM association requests, press the **Play** button icon on the **DICOM Input** toolbar. To disable Compass from accepting any new incoming association requests, press the **Pause** button icon on the **DICOM Input** toolbar. Pressing the **Pause** button icon on the **Input** toolbar does not affect any currently open associations.

To allow Compass to submit outgoing store jobs to Destinations press the **Play** button icon on the **DICOM Output** toolbar. To disable Compass from submitting any new outgoing store jobs press the **Pause** button icon on the **DICOM Output** toolbar. Pressing the **Pause** button icon on the **Output** toolbar does not affect any currently running jobs.

8.2 HL7 Input and Output

To allow Compass to accept incoming connections, press the **Play** button icon on the **HL7 Input** toolbar. This serves to start any HL7 Sources which are configured to **Start When HL7 Incoming Started**. Only sources which are started can receive incoming connections. If this **Play** button is pushed and there are sources which are not configured to **Start When HL7 Incoming Started**, these sources will still be stopped; they must be manually started to receive connections. To disable Compass from accepting any new incoming connections, press the **Pause** button icon on the **HL7 Input** toolbar. Pressing the **Pause** button icon on the **HL7 Input** toolbar will stop any HL7 Network Sources which are started, closing their open connections.

To allow Compass to submit outgoing HL7 messages jobs to Destinations press the **Play** button icon on the **HL7 Output** toolbar. This serves to start any HL7 Destinations which are configured to **Start When HL7 Outgoing Started**. Only destinations which are started can receive outgoing connections from Compass. If this **Play** button is pushed and there are destinations which are not configured to **Start When HL7 Outgoing Started**, these destinations will still be stopped; they must be manually started to receive connections from Compass. To disable Compass from submitting any new outgoing messages, press the **Pause** button icon on the **HL7 Output** toolbar. Pressing the **Pause** button icon on the **HL7 Output** toolbar will stop any HL7 Destinations which are started, closing their open connections.

Additionally, HL7 Sources and Destinations can be individually started and stopped while the HL7 Input/Output is already running. Select **View > Active HL7 Connections** to see the HL7 Connections and Nodes panel appear, and then select **Sources and Destinations** in the combo box on the right. Each Source/Destination will appear as a link, which can be clicked and stopped/started as desired.

9 Compass Cache

The **Cache** functionality requires a license. Contact Laurel Bridge Software to enable this feature on your DICOM routing Compass. Also note that use of the Compass **Cache** functionality may bring additional security considerations. See [Section 12.3 At-Rest Encryption](#) and [Appendix A: Section 2.17 Data Storage Confidentiality](#) for more details.

In a typical PACS or archive, images that are received are logically collected together in a standard Patient/Study/Series/Instance hierarchy. This organization allows users to find, for example, all of the images belonging to a single patient, or all of the studies with a given Accession Number, etc. Such a structure also allows for DICOM Query/Retrieve mechanisms (e.g. Study Root Query/Retrieve SOP Class).

While Compass is not a PACS or archive, the **Cache** capability allows for this same kind of organizational capability. When an image arrives in Compass and matches a **Rule** which is Store/Forward, and that **Rule** has the **Cache** as a **Destination**, Compass will attempt to add that image to the **Cache**. Several important aspects of the **Cache** must be understood:

- 1) There is only one copy of the image written to disk, even if the image goes both to jobs and to the **Cache**. In this way, the performance hit of adding to the **Cache** is minimized. These images are instead added to the **Cache** by reference (rather than by copy).
- 2) Because the images are added by reference, a single physical image can be in the **Cache** and in one or more **Jobs** at the same time. When such a **Job** is purged, the image remains on disk if it is in the **Cache** (and vice versa).
- 3) It is allowable to route an image to only the **Cache**.
- 4) The **Cache** feature is enabled by license, and the license controls the number of images which are allowed to be added to the **Cache**.
- 5) The **Cache** feature is intended to be exactly that: a cache. It is a relatively small, relatively ephemeral local collection/view of images for use in specific customer workflows. Again: the **Cache** is NOT an archive and is not intended to serve that purpose.

9.1 Adding to the Cache and the Penalty Box

When Compass attempts to add an image to the **Cache**, it first extracts the key fields which identify where the image belongs (Patient ID, Study Instance UID, Series Instance UID, and SOP Instance UID). This hierarchy of keys will control how the image is ingested. Under certain circumstances, the key hierarchy in an image will create a conflict. For example, an image might arrive with a Study Instance UID which **Cache** knows about, but the Patient ID in the image is different than the Patient ID associated with that Study Instance UID in the **Cache**.

If there is a conflict, or if there is a failure to ingest the image to the **Cache** for any other reason, Compass will put the image in an area called the **Penalty Box**. The **Penalty Box** is organized by transmission – all images received on a single inbound association which are put in the **Penalty Box** will be grouped together. The user can move images from the **Cache** to the **Penalty Box** and

vice versa (of course, any attempt to reingest images from the **Penalty Box** to the **Cache** may encounter the same issue which originally sent them to the **Penalty Box** when they were first transmitted).

9.2 Cache Contents and Duplicates

When an image is added to the **Cache**, in addition to the key fields mentioned above, Compass will extract a fixed collection of header fields (such as Patient Name, Study Date, Accession Number, etc.). These fields naturally belong at one of the levels in the hierarchy (for example, Study Date belongs at the Study Level). The user can configure additional custom fields to be extracted at each level in the hierarchy in the **Cache** Configuration. Because the image is written to disk as-is, and only the header fields go into the **Cache**, it is helpful to think of the **Cache** as a “view” on a collection of images.

When Compass receives a duplicate (i.e. an instance with a familiar key at one of the levels), it must decide how to handle the instance. Such handling is controlled by the **Duplication Policy**. There are currently two options.

Keep First – In this mode, the header fields provided by the first received image will be kept, and the duplicate image will provide no header fields to the **Cache**. **Keep First** is the default mode.

Keep Last – In this mode, the header fields provided by the last received image will be kept, overwriting the header fields all the way up the **Cache** hierarchy. Clearly, this mode must be used with great caution, because a later received image can impact the Cache values for an earlier received image. Consider this example:

- 1) An image is received with Patient ID “ABC”, Study Instance UID “1.2.3”, Patient Name “DOE^J”, Patient DOB “12/25/2000”, Modality “CR”, and SOP Instance UID “4.5.6”.
- 2) Later, a second image is received, belonging to a different study, but the same patient (i.e. having the same Patient ID), but the Patient DOB in this image is “1/1/2001”.

In **Keep Last** mode, the **Cache** is instructed that the Patient DOB (and other fields) for this patient are really correct, so the **Cache** will updated with the correct Patient DOB (and, perhaps, other fields). If the user then later looks at the FIRST study in the **Cache** viewer to see what the Patient DOB is, the user will find the value from the second image. Worthy of note is the fact that the first image file itself is not altered in any way; if it is sent out of the **Cache** to a **Destination**, it retains all of its original information.

9.3 Purgers

Compass provides a separate purger for the **Cache** and the **Penalty Box**, configured in the usual System configuration page. Studies will be retained in each of these areas according to the configured purge times. As noted above, purging an image from the **Cache** or **Penalty Box** will not remove that image from a **Job** (if it also in a **Job**).

9.4 Cache Query/Retrieve

Compass supports a full range of DICOM Query/Retrieve operations from the **Cache**. These operations are discussed in more detail in the Compass DICOM Conformance Claim. The Q/R operations are controlled through a few configuration settings as discussed below:

Querying Compass is controlled through the **Source** configuration. In general, when Compass receives a DICOM C-FIND DIMSE message, it must decide whether to Direct route such a message to a **Destination** or answer the message itself from the **Cache** holdings. This decision is controlled by the “**Compass Generated Responses**” section for a **Source** in the “Query/Retrieve” checkbox. If this box is checked, and a C-FIND-RQ is determined to be from a particular **Source**, that configuration value for the source will determine what happens (if it is checked, Compass will answer from the **Cache**). For security reasons, a **Source** which has the “Query/Retrieve” checkbox checked cannot be completely promiscuous (i.e. it cannot have both “Allow Any IP Address” and “Allow Any Calling Ae Title” checked).

DICOM C-MOVE requests are handled in the same fashion, using this Query/Retrieve setting. When Compass determines that it should service a C-MOVE-RQ using the **Cache**, it then extracts the Move Destination AE Title from the request. To determine which Compass **Destination** should receive the images, Compass attempts to find the **Destination** with the specified “Move AE Title”. By default, **Destinations** have this field blank (which means they may not receive images via C-MOVE-RQ); in this way, the C-MOVE support is an opt-in capability for a destination. When images are moved from the **Cache** to a **Destination**, the **Destination**’s Filters are applied to the images in process.

9.5 Cache Operations

The **Cache** and **Penalty Box** contents are viewable through the web client only (see the Web Client section of this manual for more information). The web client allows the user to search the **Cache** for patients and studies, click through the Patient/Study/Series/Instance hierarchy, look at single images, and more. Additional operations include:

Send To – This operation creates a Compass **Job** to a chosen destination and fills it with all of the images at the chosen level. A **Job** created from the Compass cache does not have an external related **Source** and hence certain fields of such a **Job**, for example **Source**, **Source IP**, and **Matched Rules**, are blank. The **Source Calling AE** is set to the value specified in the Destination node's **Compass AE Title** and the **Source Called AE** is set to the Destination node's **Destination AE Title**. The **Job** has its **Message** initially set to “Job originated from the Cache”; this **Message** is cleared when the **Job** is run. Note that such a **Job** will run according to the normal Compass **Job** rules (i.e. it might not run immediately, depending on the **Destination** schedule, current number of outbound associations, whether output is running, etc.).

Download Images – This operation downloads a zip file of all of the images at the chosen level.

Remove – This operation removes the entry of the **Cache** at the chosen level (and, by extension, all of the entries in lower levels of the cache corresponding to this entry). Note that this action will never delete image files which belong in another Compass job.

Move to Penalty Box – This operation will move all of the images under the chosen level to the Penalty Box in a single “pseudo transmission”.

Additionally, the **Penalty Box** area has the **Download Images** and **Remove** operations (analogous to those listed above), plus:

Details – This operation opens a dialog where the user can click through the images in the **Penalty Box** transmission and view them to identify what action, if any, to take with them.

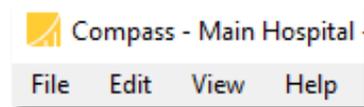
Move to Cache – This operation will attempt to reingest all of the images in a transmission (or a single image, if chosen) into the **Cache**. Note that if the images went to the **Penalty Box** due to a key collision, something must change in the **Cache** before they can be successfully reingested (e.g. the **Cache** entry causing the collision might be purged or removed).

10 Thick Client: Compass User Interface Details

10.1 Menu Bar

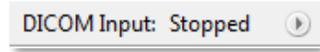
A menu bar is available at the top of the main window:

- **File > Import Configuration:** Imports an existing configuration file. Both **Input** and **Output** must be stopped in order to enable this menu item.
- **File > Export Configuration:** Exports the current configuration to a chosen location.
- **File > Exit:** Exits the program (after completing any incoming and outgoing associations).
- **Edit > DICOM Options:** Opens the main options dialog for DICOM settings and some application-wide settings
- **Edit > HL7 Options:** Opens the main options dialog for HL7 settings and some application-wide settings
- **View > DICOM View:** This check box puts the front screen in “DICOM View” mode, showing the DICOM jobs and the Active Associations Panel (if selected)
- **View > HL7 View:** This check box puts the front screen in “HL7 View” mode, showing the HL7 jobs and the HL7 Connections and Nodes Panel (if selected)
- **View > Current Application Log:** Opens the application log file.
- **View > Application Log Directory:** Opens Windows Explorer to view the Compass log file directory.
- **View > Image Reclamation Failed List:** Opens a dialog to display and operate on the list of images that could not be purged.
- **View > Inbound Association History:** Displays information regarding specific inbound associations.
- **View > DICOM Source/Destination Status:** Displays the status of each Source and Destination for DICOM
- **View > Active Associations:** Displays a table containing the current inbound and outbound associations.
- **View > DICOM Configuration Summary:** Displays a separate window with a synopsis of various DICOM configuration information for the installation.
- **View > Inbound HL7 Connection History:** Displays information regarding specific inbound HL7 connections.
- **View > HL7 Source/Destination Status:** Displays the status of each Source and Destination for HL7
- **View > Active HL7 Connections:** Displays a table containing the current inbound and outbound HL7 connections.
- **View > HL7 Configuration Summary:** Displays a separate window with a synopsis of various HL7 configuration information for the installation.
- **Help > User Manual:** Opens the Compass user manual.
- **Help > DICOM Conformance Statement:** Opens the DICOM Conformance Claim.
- **Help > Send Feedback:** Provides a mechanism to send feedback to Laurel Bridge or to request support.
- **Help > About Laurel Bridge Compass:** Displays an “About” dialog containing program and license information.

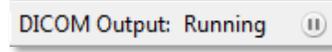


10.2 Tool Bar

The Incoming components are available in the toolbar. Pressing the **Play** button allows Compass to process incoming associations, while pressing the **Pause** button will cause Compass to no longer listen for new incoming associations. Pressing the **Pause** button does not affect currently connected associations, as they are allowed to run to completion.



The Outgoing components are available in the toolbar. Pressing the **Play** button allows Compass to process outgoing jobs, while pressing the **Pause** button will cause Compass to not start any new outgoing jobs. Pressing the **Pause** button does not affect currently running outgoing jobs, as they are allowed to run to completion.



Similarly, for HL7, the Incoming and Outgoing controls are available on the toolbar. Pressing the **Play** button for HL7 Input starts any HL7 Sources configured to **Start When HL7 Incoming Started**, causing them to listen for connections on their particular port. Pressing the **Pause** button will stop the sources, closing their open connections and their listening port. Similarly, pressing the **Play** button for HL7 Output will start any HL7 Destinations configured to **Start When HL7 Outgoing Started**. Pressing the **Pause** button will stop the destinations, closing their open connections.



10.3 DICOM Jobs

The DICOM Jobs table is displayed when the main screen is in DICOM View (**View > DICOM View**). The jobs table provides a filterable view of all the jobs in Compass. Each column is sortable and optionally displayed. Any column header can be left-clicked in order to sort by that column. Additionally, any column header can be right-clicked on in order to configure which columns are displayed. The View control allows the user to display only jobs matching the defined criteria. For example, the Jobs table can be configured to only display patient names that contain 'Carl' (see picture). Any configured filter can be saved via the **Save Current Filter** button and assigned a name for easy retrieval at a later time.

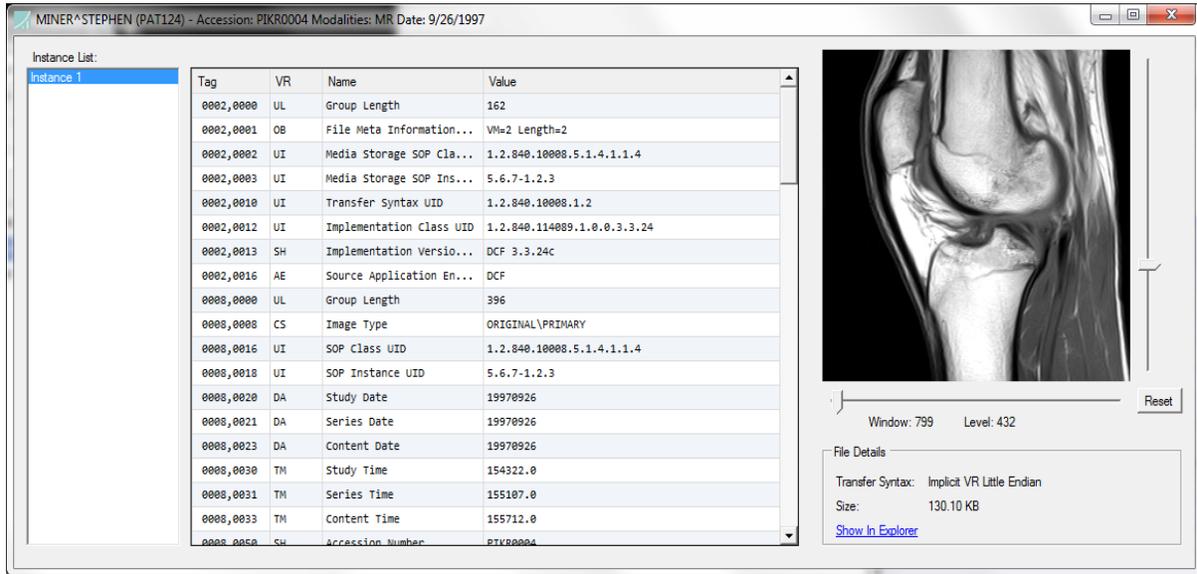
View: All Jobs

Patient Name	Patient Id	Accession #	Study Date	Modalities	Images	Destination	State
SMITH^JOE	73CB3C50	A123456	11/8/2016	US	6	Cardio PACS	Queued
DOE^JOHN	13992C5F	A1831906	11/8/2016	MR	180	Hold Queue	Hold
CARLSON^SALLY	F4D67CDE	A145649	11/8/2016	CT	200	Hold Queue	Hold

View: All Jobs where Patient Name contains 'Carl'

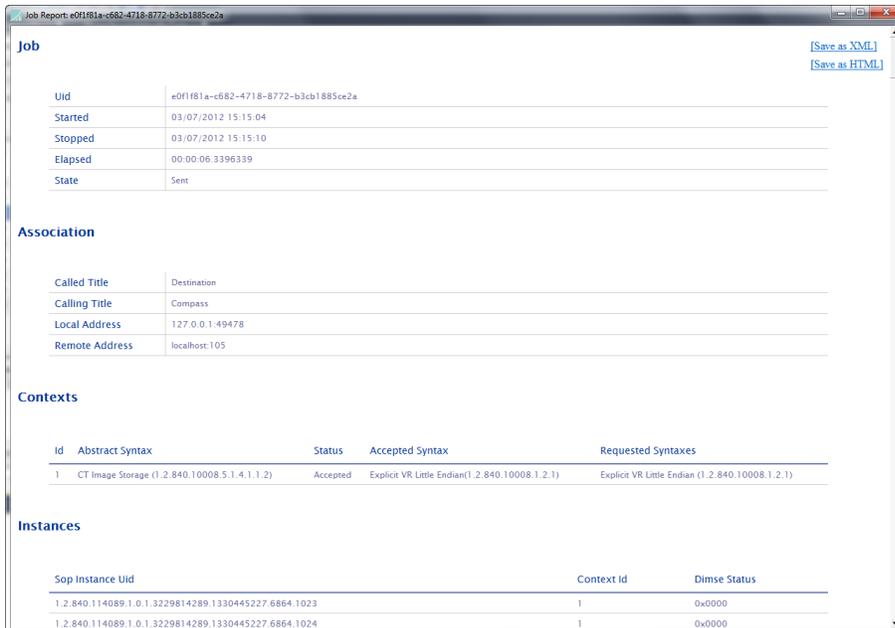
Patient Name	Patient Id	Accession #	Study Date	Modalities	Images	Destination	State
CARLSON^SALLY	F4D67CDE	A145649	11/8/2016	CT	200	Hold Queue	Hold

The jobs table also provides the ability to perform certain actions on selected jobs. A single job can be selected simply by left-clicking on the particular job row. To select a contiguous group of jobs, the user can left-click a particular job, and then left-click another job while holding down the Shift key. To select individual jobs, the user can left-click the desired jobs while holding down the Ctrl key. After selecting the desired jobs, the user can right-click to display the list of relevant actions. Possible choices are: **Details...**, **Keep and Send To**, **Remove and Send To**, **Retry**, **Retry To**, **Hold**, **Send**, **Cancel**, **Set Priority**, **Copy Job Images To...**, **View Job Report...**, and



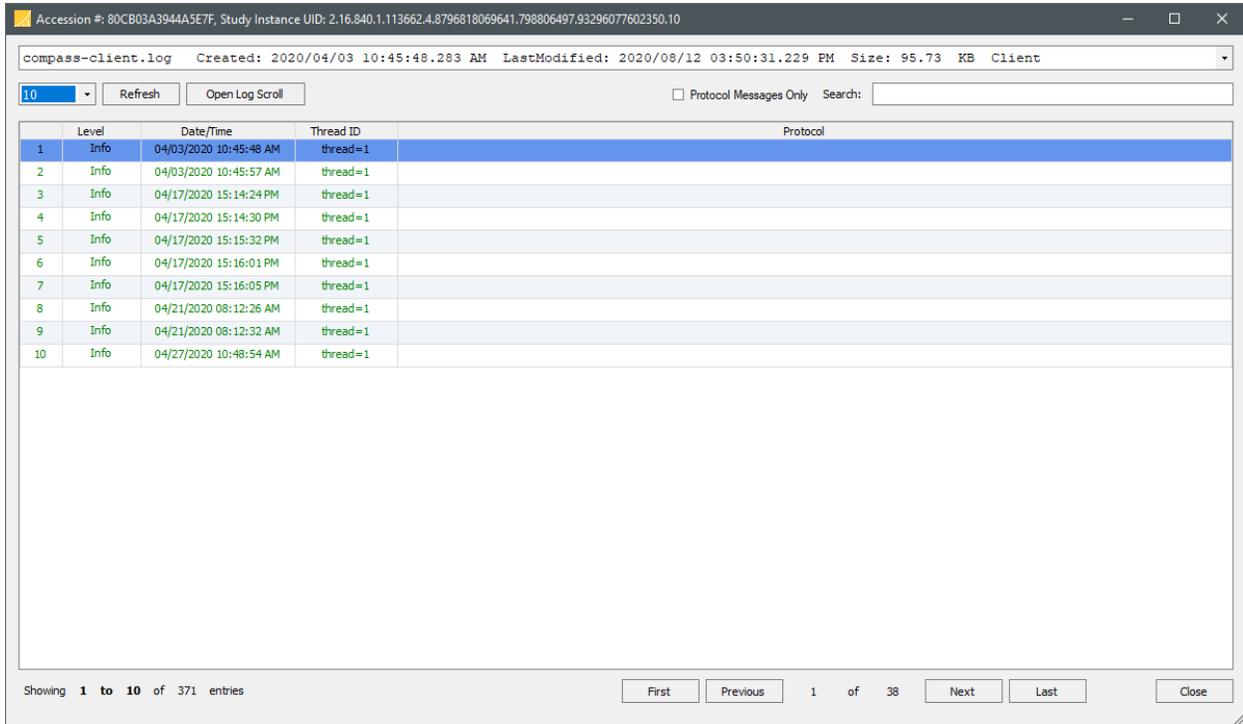
Details View

Remove. Note that if **Retry** or **Retry To** are selected, it will not cause the **Failover Destination** mechanism to kick in for the specified Destination since the number of attempts for the job has exceeded the **Max Attempts**.



10.3.1 View Logs

The **View Logs** option opens a new dialog to view the logs files that pertain to the selected job in the jobs table.



The log viewer, shown above, displays the first 10 log messages from a sample job. The drop down in the top of the form will display all of the available log messages to view for the given job. The main Compass Service log is also available in this list.

Either double click an entry in the table or click **Open Log Scroll** to view the contents of the selected log message. The selected log message will automatically appear highlighted in the **Log Scroll** after selecting an entry from the table.

The contents of the **Log Scroll** will automatically be updated to reflect the messages filtered in the table.

The **Protocol Messages Only** check box provides the ability to filter all messages not pertaining to a DICOM protocol, such as an Associate Request or Release Response message. This feature is useful, for instance, when reviewing the transactional messages from an association.

The **Search** text box allows the user to filter the selected log file given the input search text. Users may filter on a given column by using the format: COLUMN NAME:SEARCH TEXT. For instance, Level:debug will display all Debug messages for the current log file. Freeform searches are also supported, which will search the entire log file for the string entered in the search box.

10.4 HL7 Jobs

The HL7 Jobs table is displayed when the main screen is in HL7 View (**View > HL7 View**). The HL7 jobs table is substantially similar to the DICOM jobs table, only for the HL7 jobs in the system. See the section above, DICOM Jobs, for a description of the features.

10.5 Active Associations

A table containing the list of currently open inbound and outbound associations can be displayed by selecting **View > DICOM View** and checking the **Active Associations** menu item under the **View** menu. This table displays information and statistics about each open association, as well as an **Abort** button which allows the user to forcibly close the association.

	Direction	Calling AE	Called AE	Local Address	Remote Address	Start Time	Images	Receive	Send	Receive Total	Send Total
Abort	Inbound	US CART	COMPASS	127.0.0.1:111112	127.0.0.1:49386	12:29:32 PM	1614	10.29 MB/s	25.12 KB/s	102.05 MB	249.25 KB
Abort	Outbound	CLIENT	US PACS	127.0.0.1:49388	127.0.0.1:105	12:29:34 PM	489	9.61 KB/s	3.94 MB/s	75.63 KB	30.96 MB
Abort	Outbound	CLIENT	Cardio PACS	127.0.0.1:49387	127.0.0.1:106	12:29:34 PM	706	13.85 KB/s	5.67 MB/s	109.12 KB	44.67 MB

10.6 HL7 Connections and Nodes

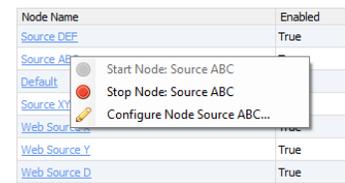
A table containing the list inbound and outbound HL7 connections and nodes can be displayed by selecting **View > HL7 View** and checking the **Active HL7 Connections** menu item under the **View** menu. This table displays information and statistics about each node, including whether it is started, stopped, or blocked. Additionally, there is a dropdown menu to let the user select only Active Connections, Source, Destinations, or Sources and Destinations. Note, for HL7 Hot Folder Sources, the Local Address column shows the configured hot folder path for that HL7 Source and the Remote Address is blank.

Node Type	Site Name	Node Name	Enabled	Node Status	Local Address	Remote Address
Source		Source_DEF	True	Running	Default:2576	
Source		Source_ABC	True	Running	Default:2577	
Destination		Default	False	Stopped		localhost:2576
Source		Source_XYZ	True	Running	Default:2575	
Source		Web Source X	True	Running	<Web>	
Source		Web Source Y	True	Running	<Web>	
Source		Web Source D	True	Running	<Web>	

To search for only certain text in nodes in the view, use the “New Filter”, type in the search text, and press the “Apply” button. If you want to group Nodes into functional groups to watch (e.g. all the nodes at one site), you can supply text for “Site Name” in their definitions, then search for that text using this view filter here.



Additionally, each of the nodes appears as a hyperlink. You can click the node and get a menu of options which apply to that node: Start the node (if it is stopped and HL7 Input is running); Stop the node (if it is running); or Configure the node (which takes you to the node’s entry in the HL7 Options dialog).



10.7 Status Bar

The status bar is displayed at the bottom of Compass' main window and displays the counts of various fields of quick interest to the user, such as current and maximum allowed inbound and outbound associations, HL7 connections, how "full" the system is with HL7 messages (as a fraction of the maximum number allowed), as well as current disk space.



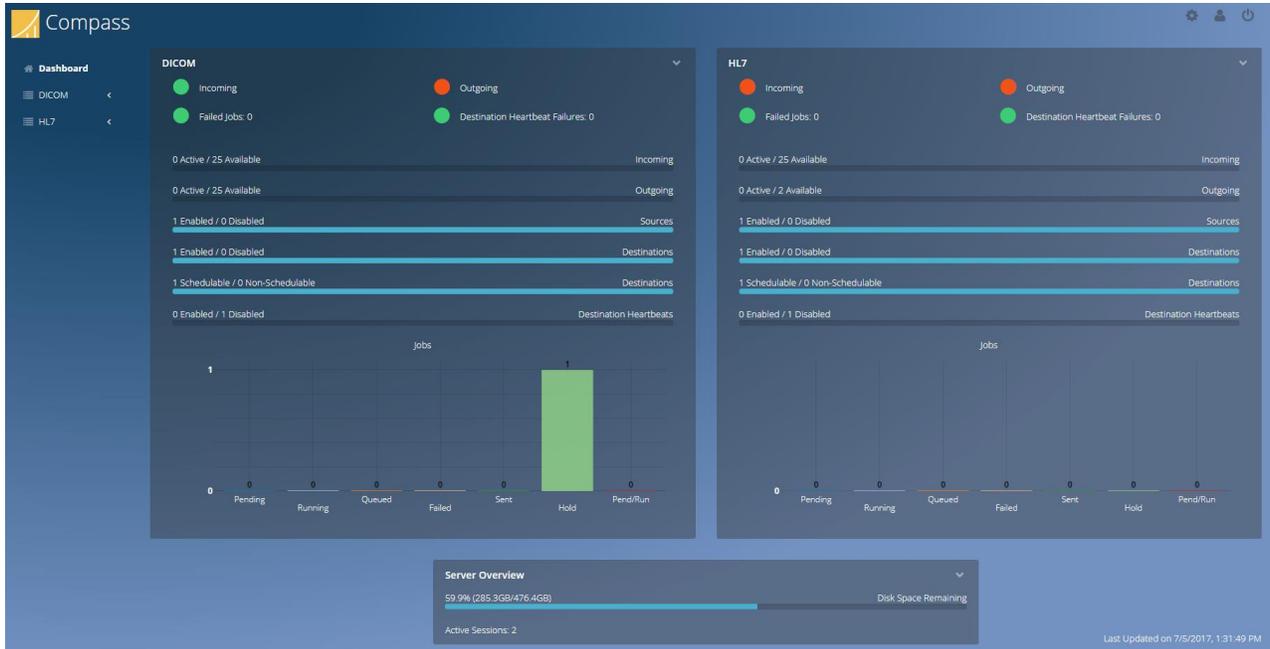
DICOM In: 0/25 DICOM Out: 0/25 HL7 In: 0/25 HL7 Out: 0/1 Jobs In View: 1 HL7 Msgs: 0% Disk Space Remaining: 80% (736.5GB/918.2GB)

11 Web Client: Compass Web Interface Details

The Compass web interface provides a mechanism for monitoring and affecting your imaging and messaging workflow.

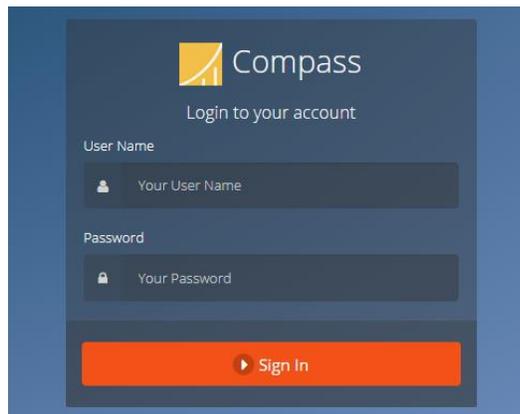
11.1 Dashboard

The main landing page is the **Status** page as pictured below. It provides a high-level overview of the Compass system. This dashboard page is the only page that does not require the user to log in.



11.2 Login

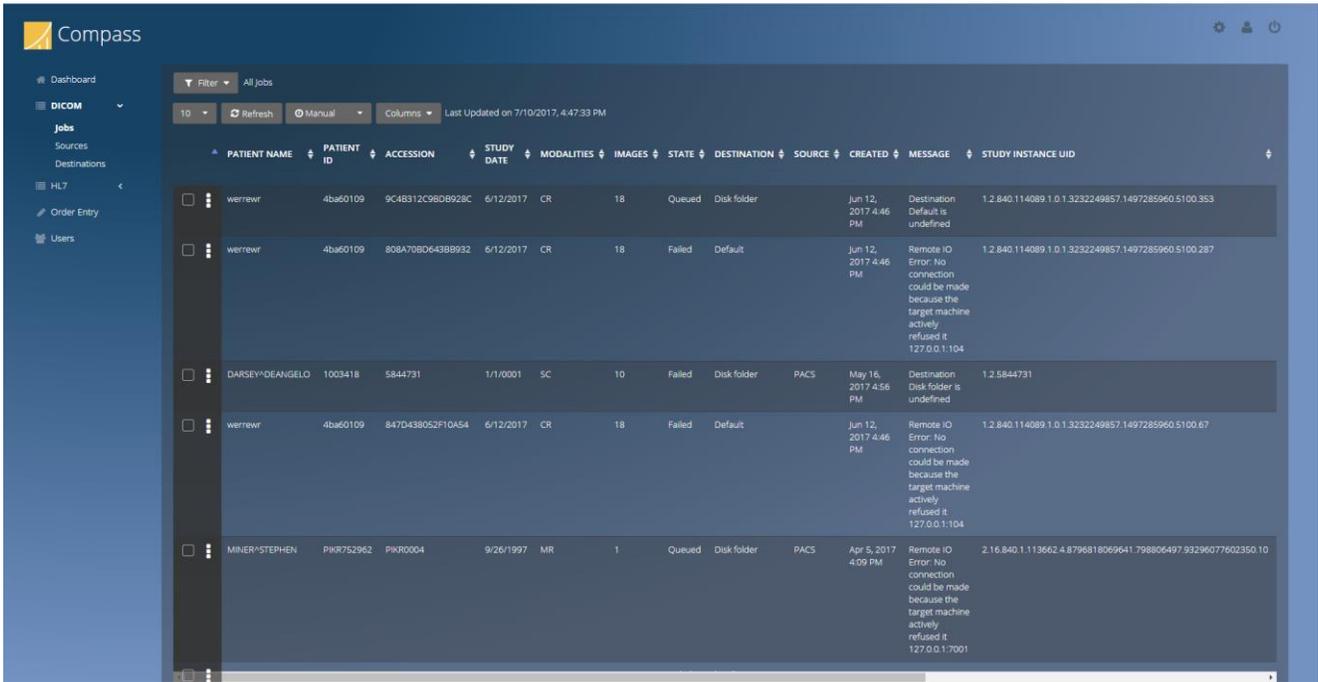
The DICOM Jobs, HL7 Jobs, and Details pages all require the user to log in. Compass allows administrative accounts to create, edit, and delete application specific usernames, passwords, and roles. Individual users may also manage their own passwords. Alternatively, Active Directory may be configured and used for user authentication and authorization. Either Compass-specific



accounts may be used, or Active Directory accounts may be used; they cannot be used simultaneously. See the **System** tab on the **Options** dialog to configure this functionality.

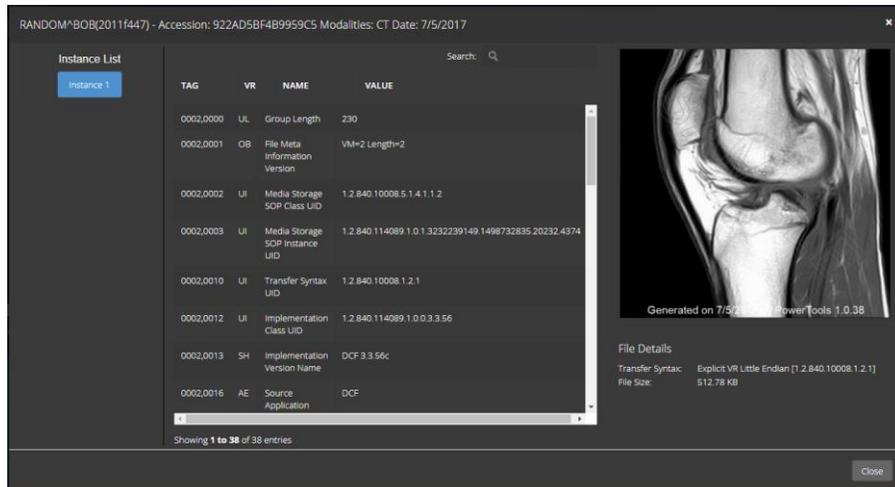
11.3 DICOM/HL7 Jobs

The **DICOM Jobs** page provides a view of the current list of Compass jobs. The table supports both filtering and sorting. The **HL7 Jobs** page works exactly the same, only for HL7 Jobs.



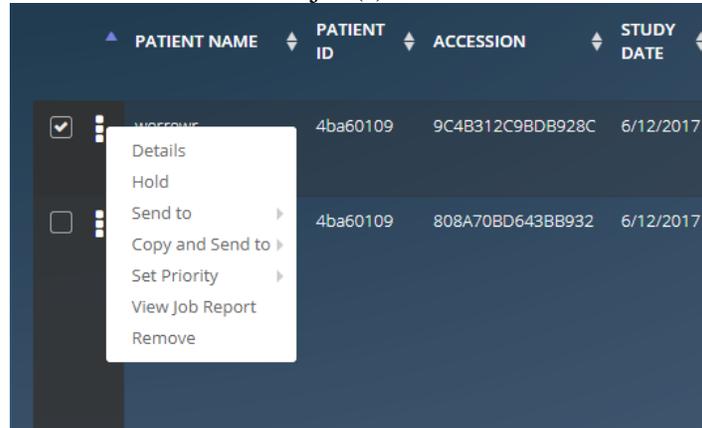
The columns can be sorted by clicking on the individual column headers.

The first column in each row of table data contains two buttons: the first is the “information” (i) button. Clicking it opens the informational details view for the job. This view provides a DICOM header dump and an image preview (when possible). You can change instances by clicking each



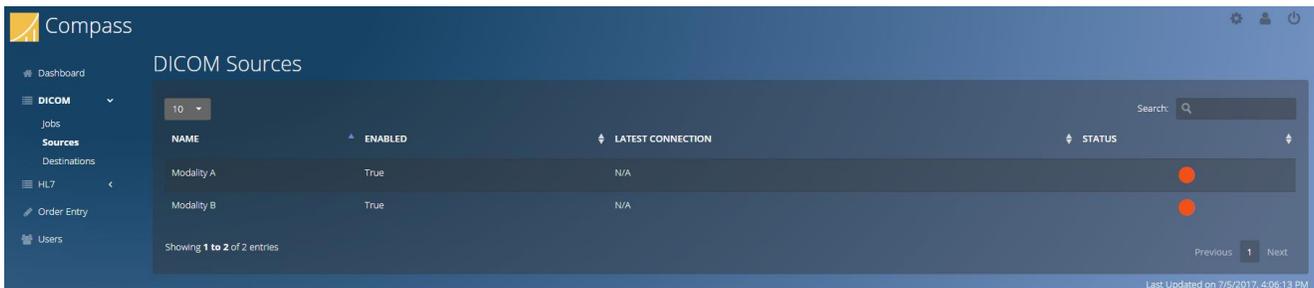
instance in the **Instance List** on the left. Clicking on the image preview will open a larger lightbox-style preview of the image scaled to the current size of your browser window.

The second button provides a context-sensitive menu which allows the user to affect the currently selected job (assuming that user has the proper permissions). Users having a role of either 'Admin' or 'User' also have the ability to modify existing jobs, such as retrying, changing the priority, copying the job to a different destination, etc. Users with a role of 'View-Only' can view, but not affect, existing jobs. Also, the list of available actions depends not only on the user's role, but also on the state of the select job(s).



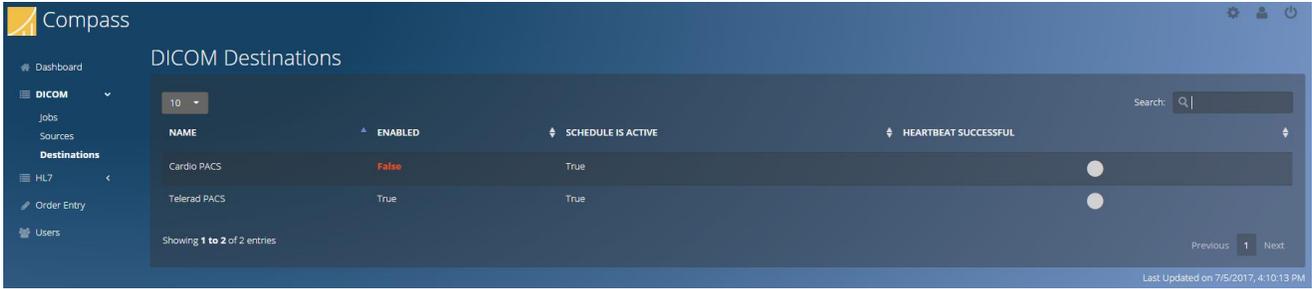
11.4 DICOM/HL7 Sources

The list of defined DICOM Sources and HL7 Sources are displayed via their respective links. The list can be filtered and/or sorted.



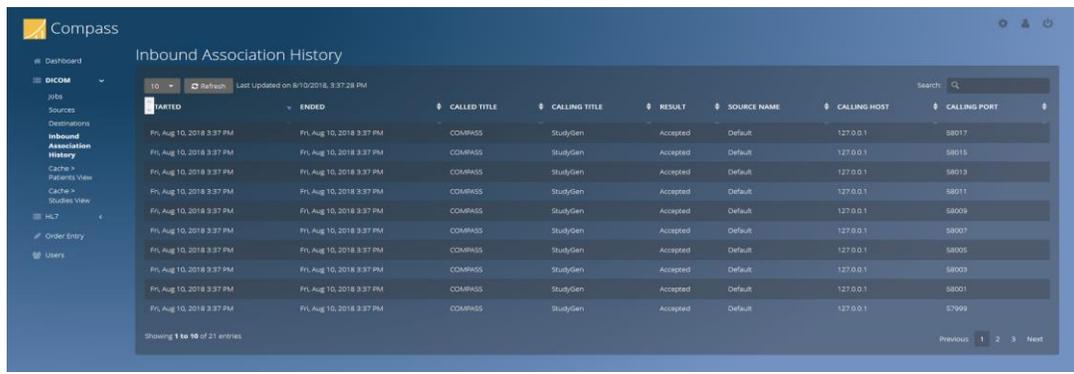
11.5 DICOM HL7 Destinations

The list of defined DICOM Destinations and HL7 Destinations are displayed via their respective links. The list can be filtered and/or sorted.



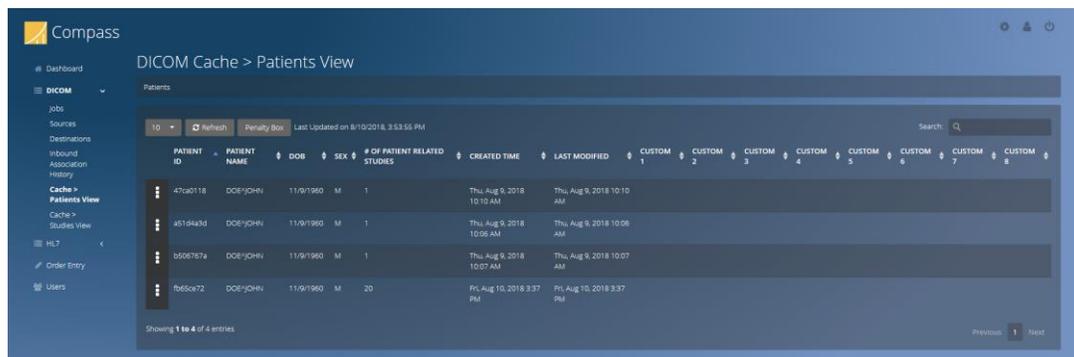
11.6 Inbound Association History

The list of inbound DICOM associations to Compass is displayed through the Inbound Association History link. This list can be filtered and sorted.



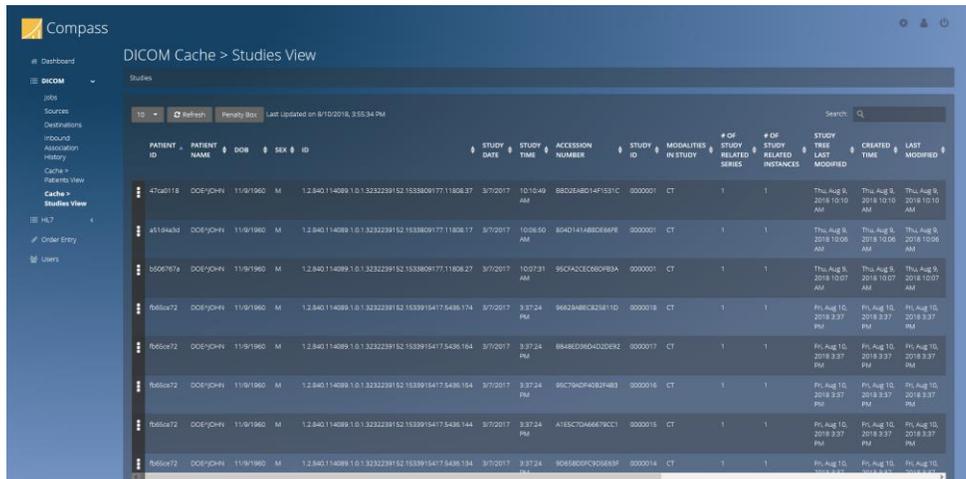
11.7 Cache Patients View

The contents of the Cache can be viewed through this navigation item, which brings the user to a list of patients, each with a unique Patient ID. The user can filter and sort this list, or perform operations described in the Cache section of this manual using the menu next to each item. The user can also navigate the Patient/Study/Series/Instance hierarchy by clicking on an item. For example, if the user clicks a patient, the view switches to all of the studies under that patient, and the navigation bar shows the patient being viewed. This approach is used at all levels of the Patient/Study/Series/Instance tree. To navigate back up the hierarchy, the navigation bar text contains links. For example, clicking the “Patients” link brings the user back to the initial patients listing.



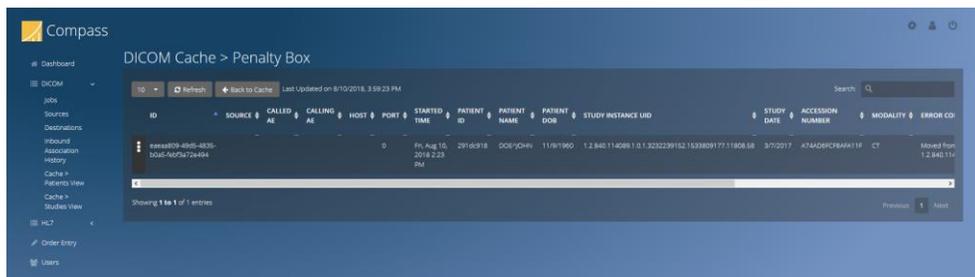
11.8 Cache Studies View

This navigation item is conceptually the same as the Cache Patients View, except that it treats all entries in the Cache as studies (as if they did not belong to a patient, even though they do). Such an approach is useful in cases where the user is search for, e.g., a study with a certain Accession Number, but the user does not know the Patient ID.

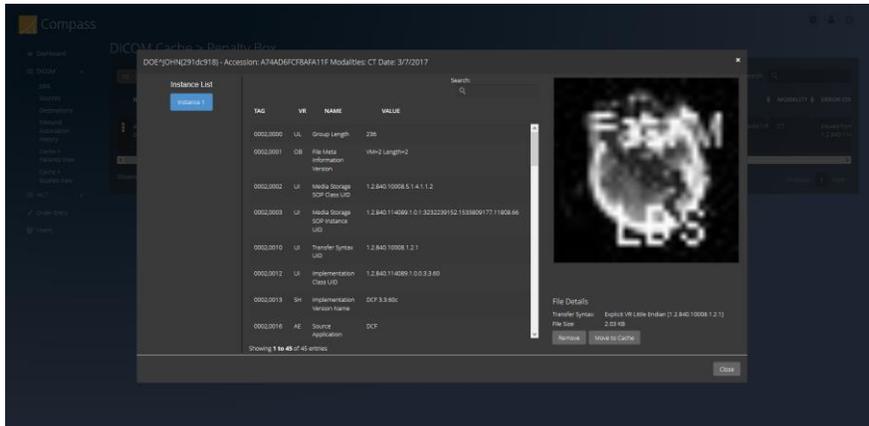


11.9 Penalty Box

The Penalty Box view is accessible from the “Penalty Box” button on either the Cache Patient View or Cache Studies View, regardless of the level (Patient/Study/Series/Instance) being viewed. The Penalty Box shows one entry per transmission which resulted in at least one “penalized” image. Normally, such a transmission is an incoming DICOM association (so, for example, if an incoming association sends 30 images to the Cache and 20 to the Penalty Box, there will be a single row in the Penalty Box corresponding to that association which has 20 images in it). If the user selects a Cache item and uses the “Move to Penalty Box” operation, there will be a “pseudo transmission” for that operation which is visible in the Penalty Box.



Clicking on a single Penalty Box transmission entry brings up a dialog where the user can navigate through the images in that transmission

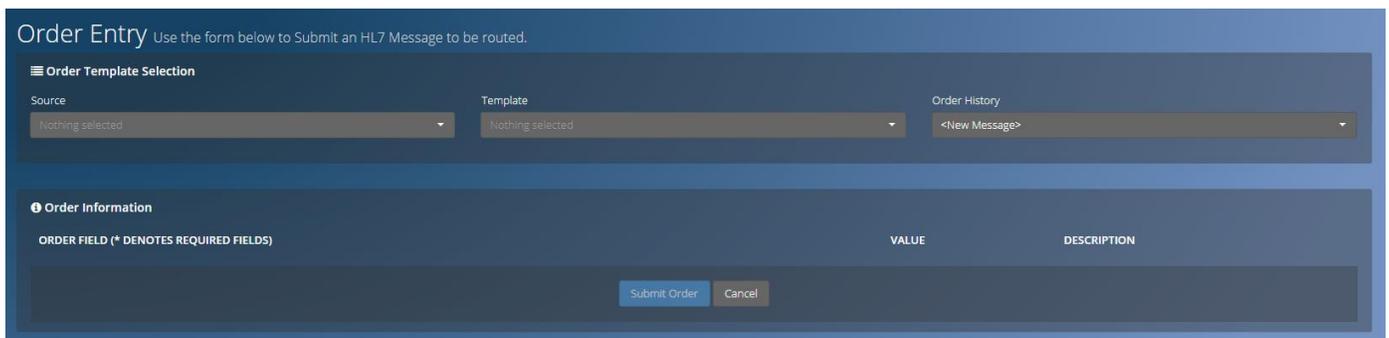


11.10 Order Entry

If the user has administrative privileges, when he logs in, there are additional tabs, including the **Order Entry** tab. This page provides the ability for the user to submit HL7 messages to Compass through the web. The user must provide a minimal configuration to use this facility. In particular:

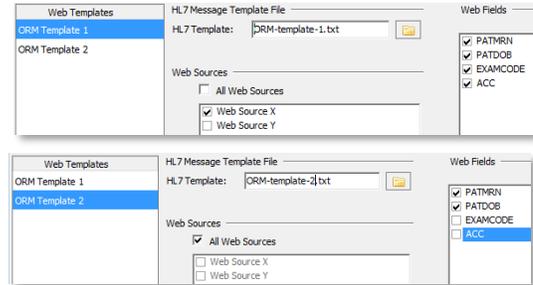
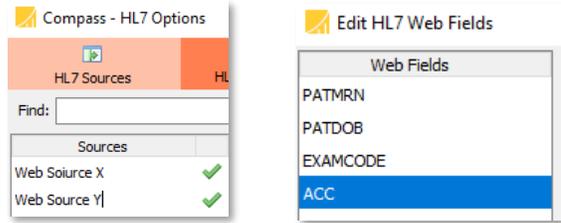
4. There must be at least one HL7 Web Source defined (**Edit > HL7 Options > Sources**), because the user will select the Web Source when filling out the form on the browser.
5. There must be at least one HL7 Web Template defined (**Edit > HL7 Options > System**), because the user will select the Web Template being used when filling out the form on the browser.
6. At least one defined HL7 Web Source must be allowed to use at least one template. Each template configuration can be configured for use with particular web sources or for all web sources.

If one of these conditions is not met, the page will show “Nothing selected” in the combo boxes:



An example configuration will help explain how the **Order Entry** feature can be used. Consider the following configurations:

- HL7 Web Sources: Two defined in **Edit > HL7 Options > HL7 Sources**. They are called “Web Source X” and “Web Source Y”.
- Fields: Four fields defined in the HL7 Web Messaging area in **Edit > HL7 Options > System**. “PATMRN” (text), “PATDOB” (date), “EXAMCODE” (list), and “ACC” (text). “EXAMCODE” points to a file of values as shown in the earlier section regarding System Configuration.
- Templates: Two templates defined in the HL7 Web Messaging area in **Edit > HL7 Options > System**. “ORM Template 1” points to the aforementioned ORM Template File, is designated specifically for use with Web Source X, and references all four of the Web Fields above. “ORM Template 2” points to a different template file, is designate for use with All Web Sources, and only references “PATMRN” and “PATDOB”.



With this configuration, when the user selects the **Order Entry** web page, the following will appear:

Note that Web Source X is selected, and both Templates are available (with the first one selected). Since this template references all four fields, there is an area for the user to supply values for those fields. The EXAMCODE field has a dropdown box for selection of values from the values file. If the user switches the selected template, the Order Information section will change to reflect the currently selected template. Also, if the user switches the selected source, only the templates which apply for that source will appear:

Order Entry Use the form below to Submit an HL7 Message to be routed.

Order Template Selection

Source: Web Source Y | Template: ORM Template 2 | Order History: <New Message>

Order Information

ORDER FIELD (* DENOTES REQUIRED FIELDS)	VALUE	DESCRIPTION
PATMRN *	<input type="text"/>	Patient Medical Record Number
PATDOB	<input type="text"/>	Patient Date of Birth

Submit Order | Cancel

The user provides the values and presses “Submit Order”. Compass will take the provided values, substitute them into the selected message template file, and submit them to the rules engine for processing, with the selected Source as the message source. If the Compass HL7 Incoming has not been started, or if the selected source is stopped, an error message will appear.

Any messages provided to Compass through the **Order Entry** page will have the supplied values remembered. These values can be reused by selected an entry in the Order History dropdown. What is displayed in this dropdown are any field values with the **Display Field For Web Message History** selected. For example: if the “ACC” field was the only **Display Field For Web Message History** field, the dropdown would contain the list of values provided by the user for the ACC field, with the most recently submitted first. If more than one field is checked as **Display Field For Web Message History**, the dropdown concatenates the values together in the order the fields exists in the field list.

11.11 Users

Viewing, editing, and deleting user accounts is accomplished via the Users link in the navigation bar:

Compass Users

10 | Refresh | Add User | Search: []

USER NAME	ROLE	EMAIL
Administrator	Admin	---
compass	View-Only	---
User3	User	---
User4	View-Only	---

Showing 1 to 4 of 4 entries | Previous | 1 | Next | Last Updated on 7/5/2017, 3:42:12 PM

Only users with a role of ‘Admin’ are allowed to create, edit, and/or delete other user accounts. Other roles are restricted to modifying their own password. [See Section 6.1.10.2 Web User Administration](#) for more detailed information and recommendations on setting up user accounts.

12 Additional Security Considerations

There are several mechanisms available within Compass to help ensure that user data is secure and protected.

12.1 Network Connections

In order to support communicating with devices that either do not support or are not configured to use encrypted connections, Compass is able to establish unencrypted connections to Sources and Destinations. However, Compass can support TLS 1.0, 1.1, 1.2 and 1.3 encrypted connections on a per-Source and per-Destination basis. It is recommended that TLS encrypted connections be used wherever possible. The option to choose an encrypted connection is available on each Source's and Destination's configuration screen in the **Options** dialog. Also, the certificate configuration can be found on the **System** tab in the **Options** dialog. See **Appendix C: Section 1.2 Configuring Secure DICOM Communication and Section 1.3 Configuring Secure HL7 Communication** for details.

12.2 Database Connections

In order to support communicating with SQL Server that has not been configured to support encrypted connections, Compass is able to establish unencrypted connections to SQL Server. However, it is recommended that encrypted connections to SQL Server be used, if possible. The option to choose an encrypted connection to SQL Server is available via the **Encrypt Connection** checkbox on the **Compass Database Configuration** dialog.

12.3 At-Rest Encryption

Compass does not encrypt the DICOM Chapter 10 files that it writes to the specified storage location. If physical access to the server(s) hosting Compass cannot be controlled, it is recommended that a full disk encryption technology (such as BitLocker or the use of self-encrypting hard drives) be used to prevent unauthorized offline access to hard drive data. See **Appendix A: Section 2.17 Data Storage Confidentiality** for more details.

13 Custom Code Extensions

There are many areas within Compass which support customization by way of custom code extensions. This section provides various examples of different extension points which are available.

13.1 DICOM Custom Filters

Custom Filter extensions are used to modify DICOM instances as they travel between Sources/Destinations and Compass. The following interfaces and examples show the contents of a custom filter file.

13.1.1 Custom Filter Action Interface

```
public interface IFilterAction
{
    public void ApplyAction(CFGGroup config, DicomSessionSettings dss,
        RelevantTagMarker tagMarker, ref DicomDataSet dds);
}
```

13.1.2 Custom Filter Action Example

```
using LaurelBridge.DCF.Configuration;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.Filters;

namespace Example
{
    public class CustomFilterAction : IFilterAction
    {
        public void ApplyAction(CFGGroup config, DicomSessionSettings dss,
            RelevantTagMarker tagMarker, ref DicomDataSet dds)
        {
            // removes the private tag (0009,1010)
            dds.RemoveElement(AttributeTag.Parse("0009,1010"));

            // better performance, visually similar
            //dds.RemoveElement(new AttributeTag(0x0009, 0x1010));

            // best performance, slightly less readable
            //dds.RemoveElement(new AttributeTag(0x00091010));
        }
    }
}
```

13.2 Custom Rule Conditions

Custom rule conditions are used to help determine whether a particular DICOM instance matches a particular routing rule. The following interface descriptions and examples show custom rule conditions.

13.2.1 DICOM Custom Rule Condition Interfaces

```
public interface IExecuteCondition
{
    bool Matches(IAssociationParameters assocParams, DicomDataSet dicomDataSet);
}

public interface IExecuteConditionEx : IExecuteCondition
{
    bool Matches(IAssociationParameters assocParams, DicomDataSet dicomDataSet,
Dictionary<string, string> parentRuleCustomConfigData);
}

public interface IExecuteConditionEx2 : IExecuteConditionEx
{
    bool Matches(IAssociationParameters assocParams, DicomDataSet dicomDataSet,
Dictionary<string, string> parentRuleCustomConfigData, string transferSyntax);
}
```

13.2.2 DICOM Custom Rule Condition Examples

```
using LaurelBridge.Compass.Core.Conditions;
using LaurelBridge.Compass.Core.Plugins;
using LaurelBridge.DCF.Dicom;

namespace Example
{
    public class CustomExecuteCondition : IExecuteCondition
    {
        public bool Matches(IAssociationParameters assocParams,
DicomDataSet dicomDataSet)
        {
            string name = dicomDataSet.GetElementStringValue(Tags.PatientName);
            name = name.Trim();
            if (name.Length % 2 == 0)
                return true;
            else
                return false;
        }
    }
}
```

13.2.3 HL7 Custom Rule Condition Interfaces

```
public interface IHL7ExecuteCondition
{
    bool Matches(IHL7ConnectionParameters connParameters, HL7Message hl7Msg,
Dictionary<string, string> parentRuleCustomConfigData);
}
```

13.2.4 HL7 Custom Rule Condition Example

```
using System.Collections.Generic;
using LaurelBridge.Compass.Core.Conditions;
using LaurelBridge.Compass.Core.Plugins;
using LaurelBridge.HL7;

namespace Example
{
    public class CustomHL7ExecuteCondition : IHL7ExecuteCondition
    {
        public bool Matches(IHL7ConnectionParameters connParameters,
            HL7Message hl7msg,
            Dictionary<string, string> parentRuleCustomConfigData)
        {
            return true;
        }
    }
}
```

13.3 Custom Job Actions

Custom job actions are run, configurably, before a DICOM job is sent, after it is sent, or after it has failed. The following interface descriptions and examples show custom job actions.

13.3.1 Custom Job Action Interfaces

```
public interface IJobAction
{
    void Action(StoreJobRecord job, IDictionary<string, string> data);
}

public interface IJobActionEx : IJobAction
{
    void Action(StoreJobRecord job, JobActionDefinition jobActionDef);
}

public interface IJobActionEx2 : IJobActionEx
{
    void Action(StoreJobRecord job, JobActionDefinition jobActionDef, CompassConfiguration
config, DestinationAvailabilityDeterminer dad);
}
```

13.3.2 Custom Job Action Example

```
using System;
using System.IO;
using System.Collections.Generic;
using LaurelBridge.Compass.Core;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.IO;

namespace Example
{
    class CustomJobAction : IJobAction
    {
        public void Action(StoreJobRecord job, IDictionary<string, string> data)
        {
            lock(typeof(CustomJobAction))
            {
                job.LoadImages();

                string dob = "NOT_PRESENT";
                string imagesDir = CompassApplication.Instance.CompassDataDirectory;
                string imageFile = job.Images[0].GetAbsolutePath(imagesDir);

                using (DicomFileInput dfi = new DicomFileInput(imageFile))
                {
                    DicomDataSet dds = dfi.ReadDataSetNoPixels();
                    try
                    {
                        // throws an exception if the tag is not present
                        dob = dds.GetElementStringValue(Tags.PatientBirthDate);
                    }
                    catch
                    {
                        // leave dob as the default value if it's not present
                    }
                }
            }
        }
    }
}
```

```
string filename = data["filename"];
string jobDescr = string.Format(@"{1} {2} {3} {4} {5}",
    job.StudyInstanceUid, job.DestinationName, dob,
    DateTime.Now, Environment.NewLine);
File.AppendAllText(filename, jobDescr);

// Generate a new accession number (for example) using any custom code
string accessionNumber = CreateNewAccessionNumber(imageFile);

// Store the value for access later on in job filters or other job actions
job.UserData["accession_number"] = accessionNumber;
}
}
private string CreateNewAccessionNumber(string imageFile)
{
    // Create a new accession number based on the data in imageFile
    return "ACC1234";
}
}
```

13.3.3 Custom HL7 Job Action Interfaces

```
public interface IHL7JobAction
{
    void Action(HL7JobRecord job, IDictionary<string, string> data);
}

public interface IHL7JobActionEx : IHL7JobAction
{
    void Action(HL7JobRecord job, JobActionDefinition jobActionDef);
}
```

13.3.4 Custom HL7 Job Action Example

```
using System;
using System.Collections.Generic;
using LaurelBridge.Compass.Core;
using LaurelBridge.HL7;

namespace Example
{
    /// <summary>
    /// Example Job Action for Changing Blank MRN to current date/time
    /// </summary>
    public class hl7OrderJobAction : IHL7JobAction
    {
        public void Action(HL7JobRecord job, IDictionary<string, string> data)
        {
            HL7Message origMessage = new HL7Message(job.FullHL7MessageString);

            // 1. Get the fields we need
            if (!origMessage.TryGetValue("MSH|10", out string controlIdString))
            {
                job.State = HL7JobRecord.JobState.Failed;
                job.FailMessage = "Message Does Not have Control ID field";
            }
        }
    }
}
```

```
        return;
    }

    // 2. Try to parse that Control ID
    int controlIdAsNumber = 0;
    try
    {
        controlIdAsNumber = Int32.Parse(controlIdString);
    }
    catch
    {
        job.State = HL7JobRecord.JobState.Failed;
        job.FailMessage = "Control ID String is not a number";
        return;
    }

    // 3. Hold jobs with odd control id string
    if (controlIdAsNumber % 2 == 1)
    {
        job.State = HL7JobRecord.JobState.Hold;
        job.FailMessage = "Control ID String is odd - holding";
        return;
    }
    handleEvenNumberedMessages(job, origMessage);
}

/// <summary>
/// Handle even numbered messages - example, set MRN to "now"
/// </summary>
/// <param name="job">the HL7 job to handle</param>
/// <param name="origMessage">original HL7 message</param>
private void handleEvenNumberedMessages(HL7JobRecord job,
    HL7Message origMessage)
{
    string newMrn = DateTime.Now.ToString("yyyyMMddhhmmss");
    origMessage.Set("PID|3", newMrn);
    job.FailMessage = "Changed MRN to new value: " + newMrn;
    job.FullHL7MessageString = origMessage.MessageString;
}
}
}
```

13.4 Custom HL7 Rule Match Actions

Custom HL7 Rule Match Actions can be run after an HL7 message received by Compass matches one of the Compass HL7 Rules. The following interface descriptions and examples show custom HL7 Rule Match actions.

13.4.1 Custom HL7 Rule Match Action Interface

```
public interface IHL7RuleMatchAction
{
    void Action(HL7RoutingRule ruleThatMatched, HL7Message receivedMessage, HL7Message
responseMessage, HL7Acceptor acceptor, HL7JobRecord job);
}
```

13.4.2 Custom HL7 Rule Match Action Example

```
using LaurelBridge.Compass.Core;
using LaurelBridge.HL7;

namespace Example
{
    public class TestRuleMatchCondition : IHL7RuleMatchAction
    {
        public void Action(HL7RoutingRule ruleThatMatched,
            HL7Message receivedMessage, HL7Message responseMessage,
            HL7Acceptor acceptor, HL7JobRecord job)
        {
            if (ruleThatMatched.CustomConfig.ContainsKey("TestRuleDataKey"))
            {
                if (job.UserData.ContainsKey("TestRuleDataKey"))
                {
                    job.UserData["TestRuleDataKey"] +=
                        ruleThatMatched.CustomConfig["TestRuleDataKey"];
                }
                else
                {
                    job.UserData["TestRuleDataKey"] =
                        ruleThatMatched.CustomConfig["TestRuleDataKey"];
                }
            }
            else
            {
                job.FailMessage = "TestRuleDataKey not found in rule: " +
                    ruleThatMatched.Name + "... failing job.";
                job.State = HL7JobRecord.JobState.Failed;
            }
        }
    }
}
```

13.5 Custom Disk Folder Serializers

Custom Disk Folder Serializers can be run when a DICOM job is to be sent to a Destination with a Transport of “DISK FOLDER”. The following is an example of a custom Disk Folder Serializer:

13.5.1 Custom Disk Folder Serializer Interfaces

```
public interface IDiskFolderJobSerializer
{
    long Deencapsulate(StoreJobRecord storeJobRecord, ImageRecord imageRecord, int
imageNumberInList, DicomDataSet dds, string dicomOutputFilePath, string
outputTransferSyntax, string destinationDiskFolderPath, bool overwriteExistingFiles);

    string CreateRelativeFilePath(StoreJobRecord storeJobRecord, ImageRecord imageRecord,
int imageCount, DicomDataSet dds);

    string SelectOutputTransferSyntax(StoreJobRecord storeJobRecord, ImageRecord
imageRecord, int imageCount, DicomDataSet dds, DicomSessionSettings sessionSettings,
string ddsEncodedTransferSyntax, StoreJobContextMode contextMode);
}
```

13.5.2 Custom Disk Folder Serializer Example

```
using System;
using LaurelBridge.Compass.Core;
using System.IO;
using LaurelBridge.Compass.Core.Utilities;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.Dicom.Store;
using LaurelBridge.DCF.Utility;

namespace Example
{
    public class AlternateDiskFolderJobSerializer : IDiskFolderJobSerializer
    {
        long IDiskFolderJobSerializer.Deencapsulate(StoreJobRecord storeJobRecord,
ImageRecord imageRecord, int imageCount, DicomDataSet dds,
string outputFilePath, string outputTransferSyntax,
string destinationDiskFolderPath, bool overwriteExistingFiles)
        {
            throw new NotImplementedException();
        }

        string IDiskFolderJobSerializer.CreateRelativeFilePath(
StoreJobRecord storeJobRecord, ImageRecord imageRecord,
int imageCount, DicomDataSet dds)
        {
            string patientPath =
                dds.GetElementStringValue(Tags.PatientID, "UNKNOWN_PATIENT")
                .Trim(MiscUtil.DicomTrimChars);
            string studyPath =
                dds.GetElementStringValue(Tags.StudyInstanceUID, "UNKNOWN_STUDY")
                .Trim(MiscUtil.DicomTrimChars);
            string seriesPath =
                dds.GetElementStringValue(Tags.SeriesInstanceUID, "UNKNOWN_SERIES")
                .Trim(MiscUtil.DicomTrimChars);
            string instancePath =
                dds.GetElementStringValue(Tags.SOPInstanceUID, "UNKNOWN_SOP_INSTANCE")
        }
    }
}
```

```
.Trim(MiscUtil.DicomTrimChars);

string relativePath =
    Path.Combine(patientPath, studyPath, seriesPath, instancePath);
return relativePath + ".dcm";
}

string IDiskFolderJobSerializer.SelectOutputTransferSyntax(
    StoreJobRecord storeJobRecord, ImageRecord imageRecord, int imageCount,
    DicomDataSet dds, DicomSessionSettings sessionSettings,
    string ddsEncodedTransferSyntax, StoreJobContextMode contextMode)
{
    // Return the default serializer's Output Transfer Syntax Selection
    return DefaultDiskFolderJobSerializer.DefaultTransferSyntaxSelection(dds,
        sessionSettings, ddsEncodedTransferSyntax, contextMode);
}
}
```

13.6 Custom File Batch Ingestor

Custom File Batch Ingestors can be run when files are received in a Compass Hot Folder Source and Non-DICOM File Support is enabled. The following is an example of a custom File Batch Ingestor:

13.6.1 Custom File Batch Ingestor Interfaces

```
public interface IFileBatchIngestor
{
    bool BeginFileBatch(IEnumerable<string> filesToProcess, string batchBaseDirectoryPath,
DicomSessionSettings sessionSettings, CFGGroup inputFilterData,
        out List<DicomElement>
dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
        out Dictionary<string, string> bonusKnowledge);

    DicomDataSet EncapsulateNonDicomAsDicom(string filePath, string
batchBaseDirectoryPath, int nonDicomInstanceNumberInBatch,
        List<DicomElement>
dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
        Dictionary<string, string> bonusKnowledge, out string transferSyntax);
}
```

13.6.2 Custom File Batch Ingestor Example

DefaultFileBatchIngestor is a publicly available class which implements IFileBatchIngestor.

```
using System.Collections.Generic;
using LaurelBridge.Compass.Core.Utilities;
using LaurelBridge.DCF.Configuration;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.Dicom.Elements;

namespace Example
{
    public class SpecialAlternateFileBatchIngestor : DefaultFileBatchIngestor
    {
        public static readonly AttributeTag BATCH_SPECIAL_TAG =
            new AttributeTag(0x0077, 0x1050);

        // Constructor
        public SpecialAlternateFileBatchIngestor() : base()
        {
        }

        public override bool BeginFileBatch(IEnumerable<string> filesToProcess,
string batchBaseDirectoryPath, DicomSessionSettings sessionSettings,
CFGGroup inputFilterData,
        out List<DicomElement>
dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
        out Dictionary<string, string> bonusKnowledge)
        {
            bool result = base.BeginFileBatch(filesToProcess, batchBaseDirectoryPath,
sessionSettings, inputFilterData,
                out dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
                out bonusKnowledge);

            // In our example implementation, we add a special bonus knowledge entry if
```

```
// there is a file in the batch with "SPECIAL" in the filename.
foreach (string file in filesToProcess)
{
    if (file.Contains("SPECIAL"))
    {
        bonusKnowledge["SPECIAL PRESENT"] = "true";
        break;
    }
}

// Return the base call result
return result;
}

public override DicomDataSet EncapsulateNonDicomAsDicom(string filePath,
string batchBaseDirectoryPath, int nonDicomInstanceNumberInBatch,
List<DicomElement>
    dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
Dictionary<string, string> bonusKnowledge, out string transferSyntax)
{
    DicomDataSet returnDataSet = base.EncapsulateNonDicomAsDicom(filePath,
batchBaseDirectoryPath, nonDicomInstanceNumberInBatch,
    dicomElementsForAllEncapsulatedFilesInBatchToReceiveFromRepresentativeBrother,
    bonusKnowledge, out transferSyntax);

    if (bonusKnowledge.ContainsKey("SPECIAL PRESENT"))
    {
        // Add a "special" key to each instance
        returnDataSet.Insert(new DicomLTElement(BATCH_SPECIAL_TAG, "XXX " +
nonDicomInstanceNumberInBatch + " XXX"));
    }

    return returnDataSet;
}
}
```

Appendix A: Compass Privacy and Security Statement

Because the Laurel Bridge Compass application is installed on hardware that is provided, configured, and controlled by the Compass customer, Laurel Bridge Software (LBS) cannot make assertions about the privacy and security of a particular Compass installation. It is up to the customer to ensure that the host Windows system onto which Compass is installed has been adequately secured and locked down. However, LBS does provide technology, tools, and guidance to assist customers in locking down their Compass installations. In the context of this appendix, the term “Compass customer” refers to the administrators for the host hardware system and for the Compass application.

Section 3 - GDPR Notes found below contains comments regarding the European Union’s (EU’s) GDPR - General Data Protection Regulation.

An overview of the Compass application privacy and security features is given in the sections below, roughly following the format given in the HIMSS/NEMA Standard HN 1-2013, “Manufacturer Disclosure Statement for Medical Device Security”, or MDS2 for short. For more details about this form or to download it, see

<https://www.nema.org/Standards/Pages/Manufacturer-Disclosure-Statement-for-Medical-Device-Security.aspx>

The headers in the following sections map directly to the headers in the MDS2 document. The Compass MDS2 document is included with the installation files and is available for a particular release version upon request from LBS.

1 Management of Private Data

The Laurel Bridge Compass application acts as a router for both DICOM messages and HL7 version 2.x messages, both of which may contain protected health information (PHI). Compass can route these messages from one or more sources to one or more destinations. Consequently, Compass can ingest, store, display, and transmit PHI. However, since the PHI only resides in Compass temporarily, Compass is not considered a primary repository of electronic health record (EHR) or electronic medical record (EMR) data, and thus is not maintaining part of the designated record set (as defined by HIPAA). Also, the Compass application and the data it stores and manages is entirely resident within the customer premises (i.e., no part of the application or its data is cloud-hosted or hosted by LBS).

1.1 Types of PHI Maintained

Because Compass is able to handle both DICOM and HL7 messages, it potentially transports and caches the following types of PHI:

- Patient demographic information
- Patient medical record information
- Patient diagnostic and therapeutic information (including diagnostic images)
- Patient financial information

1.2 Persistence of Private Data

Compass maintains PHI both temporarily in memory (while running) and on disk (persistent storage). PHI may be found in data transmitted or cached by the application, and in log files generated during use of the application. Data can be imported from or exported to other medical systems via network-mounted hot folders or removable media, but these features must be explicitly configured by the Compass customer, and their use remains under the full control of the customer. Available security features to protect PHI when at rest are described below and, in more detail, elsewhere in this Compass User Manual.

Note: Due to the sensitive nature of the PHI that Compass handles, the only non-destructive and completely safe way to decommission a (non-virtual) computer system on which a production Compass application has been running is to wipe the hard drive clean using a suitable hard drive wiping application. For self-encrypting drives, changing or overwriting the encryption key(s) may be sufficient.

1.3 Transmission of Private Data

PHI can be transmitted or received over the network via DICOM, HL7, or other messages. The ability to configure and control the behavior of this functionality is under the full control of the Compass customer, and the use of these features remains under the full control of the customer. Available security features to protect PHI when in transit are described below and, in more detail, elsewhere in this Compass User Manual.

1.4 Payment Card Industry (PCI) Data Security Standard

Because Compass does not process any patient billing transactions, it is not subject to the requirements of the Payment Card Industry (PCI) Data Security Standard.

2 Security Capabilities

The Laurel Bridge Compass application is comprised of three parts:

1. **Compass Service**, which runs as a Windows Service
2. **Compass Client**, a Windows Forms-based (UI) application which is used to configure the Compass Service and monitor the jobs of which Compass is aware
3. **Compass Web**, an optional web interface that allows configured web users to monitor and manage jobs

Note that the Compass Client does not need to be running for the Compass Service to run, accept, process, and send jobs.

The following sections briefly describe available security features of the Compass application. For more details, see the Compass User Manual.

2.1 Automatic Logoff

The Compass Client (the UI used to configure the Compass application) does not automatically time out or log a Windows user off, so LBS recommends that the Windows password-protected, inactivity-activated screen lockout functionality be enabled for all users with Compass privileges on the host Windows system, to help avoid unintended administrative access by unauthorized users.

The Compass Web interface can be configured to automatically log off Compass users in a configurable number of minutes. The default timeout is 10 minutes, and the timeout can be configured to any value from 1 minute to 65536 minutes. Note that enabling the web auto-refresh functionality on the status screen disables the web user auto-logoff.

2.2 Audit Controls

Compass can be configured to send DICOM PS3.15 Appendix A.5 (“Audit Trail Message Format Profile”) audit messages to a syslog server (such as **syslog-ng** or **nsyslog**). Messages can be sent via the TLS (recommended), UDP, or TCP protocols, and all messages include the user ID of the user performing the action as well as a date/time stamp.

The following types of audit trail messages can be enabled/disabled independently:

- **Application Start/Stop** – Logs when an application is started/stopped.
- **Software Configuration** – Logs when changes are made to the software configuration.
- **DICOM Instance Network Transfer** – Logs when DICOM instances are transmitted via the network.
- **DICOM Instance Import/Export** – Logs when DICOM instances are imported/exported.
- **User/Security Alerts** – Logs when web user or security alerts occur. These include events such as web user logon/logoff, web user addition/removal, web user password/role changes, and manual modifications of DICOM or HL7 jobs.

The following DICOM PS3.15 Appendix A.5 audit trail message types are supported by Compass:

- **Application Activity**
 - Application Start

- Application Stop
- **Audit Log Used**
- **Begin Transferring DICOM Instances**
- **Data Export** (optional configuration)
- **Data Import** (optional configuration)
- **DICOM Instances Accessed**
- **DICOM Instances Transferred**
- **Security Alert**
 - Security Configuration
 - Software Configuration
 - Use of Restricted Function
 - User Security Attributes Changed
- **User Authentication**
 - Login
 - Logout

2.3 User Authorization

Windows Authentication (whether locally-administered or domain-based) is used to control access to the Compass Client (and thus the ability to configure the Compass Service). The Compass Client can be configured to require administrative privileges on the host Windows system in order to run. Doing so means that only Windows users with administrative privileges have administrative control over the Compass application. This configuration also locks down the Compass data directory so that only users with administrative privileges can access the temporary DICOM images and Compass configuration and log files.

If the Compass customer chooses to allow the Compass Client to be run by normal users, then it is up to the customer to lock down administrative control of Compass (e.g., by using a custom Windows user-level group).

The Compass Client can also be configured to require entry of the valid password for the currently-logged-in Windows user. This prevents unauthorized users from accessing the Compass Client in the event that the host Windows system console is not kept locked at all times when not in use (i.e., if not using a short Windows auto-logout timeout).

The Compass Web users can either be locally administered (by the Compass Web module), or they can be administered using LDAP / Active Directory. This is done by the Compass customer configuring one or more Active Directory groups for each of following built-in web user roles:

- Admin user
- Regular user
- View-only user

2.4 Security Configuration

The Compass customer has full control over and responsibility for the security of Compass, both through the ability to lock down the Windows system on which Compass is installed, as well as through the ability to configure the security features built into the Compass application. Extensive information about how to do this is found in this Compass User Manual.

2.5 Security Updates

The Compass customer has full control over the installation of Windows security updates, as well as over the installation of any Compass application updates.

2.6 De-Identification of PHI

Compass does support the ability to configure de-identification of PHI. However, due to the consequences this can have on the usability of the DICOM and HL7 messages, this is typically only configured when sending images containing PHI to external organizations (such as organizations which may use the images for academic or publication purposes). When sending PHI over the public internet to / from satellite locations, the use of TLS encryption or an encrypted VLAN is the way recommended by the DICOM Standard to protect the confidentiality of PHI in transit.

2.7 Backup and Restore

The Compass customer has full responsibility to both install and maintain the SQL Server database which provides the backing store for the Compass jobs. As such, the customer is also responsible for providing backup and restore capabilities for the SQL Server database. Microsoft provides an extensive set of SQL Server backup, restore, and replication technologies.

2.8 Emergency Access

Since the Compass customer has full control over the installation and configuration of both the host system and the Compass application itself, it is up to the customer to provide a means of emergency access (“break-glass” feature) by maintaining alternate access to administrative credentials for the systems involved.

2.9 Data Integrity and Authenticity

Since one of the primary functions of Compass is to modify and route DICOM and HL7 messages, it is simply not practical to implement a mechanism whereby alteration of data can be detected. Instead, the following techniques can be used to control and track data modifications:

- Use Audit Trail logging to record any access to or modification of data.
- Use Windows Authentication to ensure that unauthorized Windows users cannot access the host Windows system on which Compass is installed.
- Use Compass Web authentication (either locally-administered or based on Windows Authentication) to ensure that unauthorized web users cannot access the Compass data remotely.
- Use TLS encryption on the network connections used by the system to ensure privacy, node authentication, and protection against man-in-the-middle (MITM) attacks.
- Use full disk encryption technology (such as BitLocker or self-encrypting hard drives) if physical access to the server(s) hosting Compass cannot be controlled, See [Appendix A: Section 2.17 Data Storage Confidentiality](#) for more details.

Compass does not currently use explicit error detection on data at rest, but rather depends on the built-in ECC error detection and correction technology provided by modern hard drives (as supported by Windows). If data redundancy is desired, LBS recommends the use of RAID data storage technology for both the SQL Server database repository and for the DICOM image cache.

2.10 Malware Protection

Since the Compass customer has full control over the installation and configuration of both the host Windows system and the Compass application itself, it is up to the customer to install and maintain malware protection technology. Compass itself should be unaffected by the use of such technology (beyond the obvious potential impact to system performance that can occur when using anti-virus software). For network router performance, it is generally recommended that antivirus checking be turned off for the temporary directories and SQL data directories used by Compass.

2.11 Node Authentication

Node authentication (the ability to confirm the identity of both the sender and receiver of DICOM and HL7 data) can be implemented using TLS protocols on all network connections. Compass supports TLS versions 1.0, 1.1, 1.2 and 1.3 as both client and server. TLS must be enabled separately on both DICOM and HL7 inputs and outputs, as well as on the Compass Web interface. More details about how to do this and further security details can be found elsewhere in this Compass User Manual.

2.12 Person Authentication

As mentioned earlier, user authentication for the host Windows system can be controlled locally, using a domain with single sign-on technology such as LDAP / Active Directory. User authentication for web interface users can also be controlled either locally or using LDAP/AD.

2.12.1 Local Web User Administration

If you elect to administer web users locally, then there are no limits placed on the number of user accounts that can be created. Customers can and should immediately change default passwords during the installation process (there are only two default accounts, “administrator” and a view-only user “compass”). Passwords must be a minimum of 8 characters long and must contain both uppercase and lowercase letters. Optionally, a high-security password mode can be enabled, which requires that passwords be a minimum of 12 characters long and must contain numeric digits, in addition to uppercase and lowercase letters. Shared user IDs can be used, but the default behavior is to only allow a user to log on from a single computer at a time. Local users’ passwords cannot currently be configured to expire.

2.12.2 Single Sign-On (LDAP/AD) Web User Administration

When web users are administered via a single sign-on technology such as LDAP/AD (recommended), the rules regarding users and passwords are up to the single sign-on technology. Active Directory allows for the configuration of password complexity and expiration rules, account locking, centralized account administration, etc.

2.13 Physical Locks

Since the Compass customer owns and has full control over the host Windows system on which Compass is installed, it is up to the customer to maintain the physical security of the host system.

2.14 Device Life Cycle Roadmap

As detailed in the User Manual minimum specifications, the Compass application currently supports the following Windows operating systems:

- Windows 10
- Windows Server 2012 R2
- Windows Server 2016

LBS intends to support each of these operating systems up until their respective end-of-extended-support dates.

In addition, the Compass application has the following software dependencies:

- SQL Server (can be SQL Server 2012 x64 or newer)
- SQL Server Management Studio
- .NET Framework 4.8 (or later)

2.15 System and Application Hardening

Since the Compass customer provides, configures, owns, and has full control over the host system on which Compass is installed, it is up to the customer to perform system hardening, as well as to configure the Compass application for the desired level of application hardening. More details about hardening of the host Windows system and the Compass application can be found elsewhere in this Compass User Manual.

Some specific application hardening techniques that are supported by and/or implemented in Compass include:

- Use of Authenticode digital signatures (currently SHA256) for all LBS executables and DLLs
- Support for TLS encryption for data in transit
- Provision of instructions for how to lock down the TLS protocols and ciphers, which affects both the Compass Web interface, as well as any DICOM or HL7 connections which are configured to use TLS encryption (see User Manual, Appendix C: Communicating Securely with Compass)
- Support for single sign on (Windows Authentication / Active Directory)
- Support for PHI anonymization for exported data (can be reversible or non-reversible)

The implementation of the following lockdown techniques on the host Windows system is the responsibility of the Compass customer:

- Disabling of unnecessary Windows accounts
- Disabling of unnecessary open network ports (e.g., telnet, ftp, etc.)
- Removal of any unnecessary off-the-shelf applications

- Enabling of Windows password-protected, inactivity-activated screen lockout
- Disabling of the ability to boot from removable media (if physical access to the host Windows system cannot be controlled)
- Enabling of BitLocker or other at-rest, full-disk encryption technologies (if physical access to the server(s) hosting Compass cannot be controlled) – see [Appendix A: Section 2.17 Data Storage Confidentiality](#) for more details.
- Enabling of SQL Server encryption (especially if the database resides on a different, unencrypted system)

2.16 Security Guidance

The security-related features of the Compass application are described in detail in this Compass User Manual.

2.17 Data Storage Confidentiality

Compass does not encrypt data while at rest on the hard drive(s). PHI is stored both in the SQL Server database, as well as in the cached data files. If at-rest encryption of PHI is deemed necessary (e.g., if physical access to the host Windows system cannot be controlled), we recommend the use of a full disk encryption technology such as BitLocker or the use of self-encrypting drives. SQL Server at-rest encryption technologies such as Transparent Data Encryption (TDE) may also be necessary if the SQL Server database is resident on a different (unencrypted) system. Compass does support encrypted SQL Server connections, and their use is highly recommended in the case of SQL Server instances accessed over a network.

2.18 Data Transmission Confidentiality

Compass can be configured to encrypt data in transit (using TLS), which will protect the data against interception by unauthorized parties. It can also be configured to use reversible de-identification, which will encrypt PHI in such a manner that it can be decrypted by the authorized receiver. And as mentioned above, Compass supports encrypted SQL Server connections, and LBS highly recommends using them in the case of SQL Server instances accessed over a network.

2.19 Data Transmission Integrity

TLS encryption also protects the data against any attempt to modify the data during transmission (i.e., via man-in-the-middle attacks). Compass will only transmit data to destinations that have been explicitly configured within the application by the customer.

2.20 Other Security Considerations

Compass can be serviced remotely by LBS only with the express permission of the Compass customer, as access to the host system onto which Compass is installed is completely controlled by the customer. Compass does not contain any service backdoors, nor does it contain any secret service accounts. All LBS access to an installed Compass application must be explicitly enabled/allowed by the customer using standard Windows secure remote access technologies.

The following port numbers are the defaults used by the Compass application. Note that these can all be changed by the Compass customer, if so desired.

- DICOM input port = **11112** (**2762** if using TLS)
- HL7 input port = **2575** (same if using TLS)
- HTTP port = **10400** (**10401** if using HTTPS)
- Lighthouse communication port = **54500**

3 GDPR Notes

The European Union's (EU) General Data Protection Regulation (GDPR) is a refresh of Europe's data-protection laws that harmonizes statutes across the 28 EU member states; it became effective May 25, 2018. GDPR is a law that applies to any organization doing business in the EU or with EU-based clients. It is up to the Laurel Bridge application customer to ensure that they manage the Compass application and the medical imaging data processed by it in a way that is conformant to their GDPR policies and practices.

The content in this appendix describes the relevant security and privacy information associated with this application. Relative to the GDPR some key points to remember are:

- The Laurel Bridge application is installed on virtual or physical systems that are provided, configured, and controlled by the customer, therefore Laurel Bridge Software (LBS) cannot make assertions about the privacy and security of a particular installation.
- It is up to the customer to ensure that the customer's host systems on which the application components are installed have been adequately secured.
- By virtue of using this application, Laurel Bridge Software receives no private data from the customer or the customer's clients; data remains with and under the control of the customer.
- The application does not maintain a designated record set and is not a primary repository of electronic health record (EHR) or electronic medical record (EMR) data. Data processed and tracked by the application is transient and purged after a user-configurable period of time.
- Section 1 in this appendix, Management of Private Data, describes private data that may be processed by the application and which may be relevant to the customer's GDPR compliance activities.
- Log files may possibly contain private data associated with the medical imaging data being processed. Such files should be handled in a way that is compliant with the customer's data retention and privacy policies.

Appendix B: Compass FAQs

1 How can a SOP class be added to the default list?

On the Compass GUI Main Dialog window, stop the Input and Output from the Main Dialog.

Go to the Edit→Options→System-Tab→Edit-Supported-Sop-Classes, then add the desired SOP classes (including private SOP classes) to the list of supported SOP classes.

Make sure you click the plus ‘+’ button to update the list.

Once so added, Compass will accept them and then you should be able to make Routing Rules based on the SOP class UID field and route those DICOM objects where ever you want including to the Hold queue or just discard them.

2 How can I distinguish between my non-DICOM jobs in the user interface?

If Compass is configured to ingest non-DICOM files via a Hot Folder Source, these files will be converted into DICOM representations of the file content. In many cases, these DICOM representations will not include the typical information used to identify Compass jobs (Patient ID, Patient Name, Accession Number, etc.).

However, there is one simple way to tell which job represents which ingested batch of files. When Compass ingests these files, it creates (among other things) a Study Description for the batch. This Study Description will indicate the name of the first file Compass discovered at import, as well as the number of files in the batch. So, for example, the Study Description might be:

Myfile.pdf ... [1 file]

or:

Mydir\myfile.docx ... [15 files]

Because this is a Study Description, Compass can index the value into its jobs database using the Custom Column feature. To achieve this end:

- a) Select Edit-> DICOM Options-> System
- b) Under “Custom Table Columns”, pick one of the Columns, enter “Study Desc.” (or whatever label is desired), and then select the “Study Description” tag using the pulldown menu.



From this point on, when Compass receives any job, you can view the Study Description in the Jobs viewer on the main User Interface (you will need to scroll to the right to see it initially).

3 How do I remove private tags from a DICOM data set?

You will need a custom filter to remove private tags.

To be able to properly identify the private tags, first you need to create a private data dictionary, which is described in the section on using the Composer Action.

Once you have the private dictionary, then the following block of code would remove two private tags:

```
using LaurelBridge.DCF.Configuration;
using LaurelBridge.DCF.Dicom;
using LaurelBridge.DCF.Filters;

namespace Example
{
    public class RemovePrivateTag : IFilterAction
    {
        public void ApplyAction(CFGGroup config, DicomSessionSettings dss,
            RelevantTagMarker tagMarker, ref DicomDataSet dds)
        {
            // Need the following line to pull in the elements trailing the
            // pixel data, if the private tags come after the pixel data.
            //dds.ExpandStreamingModeData(true);

            // Example private tags to be removed
            dds.RemoveElement(new AttributeTag(0x07A1, 0x1050));
            dds.RemoveElement(new AttributeTag(0x07A3, 0x1014));
        }
    }
}
```

Appendix C: Communicating Securely with Compass

1 Secure DICOM and HL7 Communication with Compass

The Compass application may be configured to communicate securely with another device by enabling the appropriate TLS (Transport Layer Security) options. Typically, this feature is used to enable secure DICOM communications between two peer copies of Compass.

1.1 Overview

Compass supports the BCP195 TLS Secure Transport Connection Profile (See DICOM PS3.15 Security and System Management Profiles, Appendix B.9) for authentication and encryption of communication between it and other DICOM clients and servers. Compass supports TLS version 1.2 as required by this profile.

1.2 Configuring Secure DICOM Communication

For secure DICOM communication to another DICOM application, one should select the **Use TLS** option under the **Advanced** section of the DICOM **Destinations** pane (by selecting one or more TLS versions to support). See [Section 4.6 Creating a Listener](#) for more details. For secure DICOM communication from another DICOM application, one should configure one or more encrypted listeners under the **Listeners** tab by enabling one of the **Use TLS** for the listener. See [Section 6.1.2 DICOM Incoming](#) for more details.

To properly configure incoming DICOM TLS connections, on the DICOM **System** pane, the **Listen Port**, **TLS Certificate** path and certificate **Password** must be set. The **Listen Port** default is port 2762, as recommended by the standard; it is the port on which Compass will receive TLS encrypted communications (e.g., DICOM associations). The **TLS Certificate** path should be set to the file system location of the certificate that Compass should present for identification to clients. It is suggested that the certificate be a standard PKCS#12 certificate and it must contain an exportable private key. The **Password** must be set to the password for the private key in the certificate. Note: Using a certificate format that does not password-protect the private key allows the password setting to be ignored, but we highly recommend keeping private keys password-protected. This appendix does not describe the procedures that are required to obtain TLS certificates.

This configured certificate information will be provided to any DICOM TLS client that connects to Compass on the TLS listen port. This configured certificate information can also be provided to any server that Compass connects to, if configured to do so. All communication over this connection is then encrypted using the exchanged certificate(s). Note that this TLS-level authentication is in addition to the DICOM-level authentication that Compass provides based on DICOM AE titles and IP addresses (which are used to determine whether or not to allow a DICOM client to open a DICOM association to Compass).

When TLS is enabled on a **Destination** (i.e., Compass is the client), Compass will use the TLS secure communication mechanism to request the server's certificate when it makes a connection to the server. It will then authenticate the server and encrypt all communications with the server.

Compass can optionally be configured to send its configured certificate information when making a connection, which allows the server to authenticate the client as well.

When TLS is enabled for a **Listener** (i.e., Compass is the server), Compass will send its configured certificate information to the client when it makes a connection to an encrypted listener. The client uses this information to authenticate and encrypt all communication with Compass. Compass can optionally be configured to require the client's certificate when making a connection to it. In this case, if the client does not send its certificate, the connection will be refused. The default is to allow missing client certificates (no client authentication), which is similar to how web browsers work.

On both the client and server sides, Compass may optionally be configured to accept self-signed certificates or to ignore certificate name mismatch errors. These options are primarily intended for testing purposes, and for optimal security, we recommend that these be left unchecked. TLS authentication certificates should ideally be obtained from a trustworthy TLS certificate authority (CA). If this is not feasible, a self-signed certificate can be generated for a local computer (see **Appendix E**: for instructions), manually (and securely) copied to the remote computer, and installed into the certificate store on that computer as a trusted root certificate.

See DICOM PS3.15 Security and System Management Profiles, Appendix B.9, for a further description of the Basic TLS Secure Transport Connection Profile. This document may be found at: <http://medical.nema.org/standard.html>.

1.3 Configuring Secure HL7 Communication

For secure HL7 communication to another HL7 application, one should select the **Use TLS** option under the **Advanced** section of the HL7 **Destinations** pane (by selecting one or more TLS versions to support). See **Section 5.2.3 Advanced Settings** for more details. For secure HL7 communication from another HL7 application, one should select the **Use TLS** option under the **Compass Listen Configuration for Source** section of the **HL7 Sources** pane (by selecting one or more TLS versions to support). See **Section 5.1.1.1 Settings** for more details.

When TLS is enabled on an **HL7 Destination** (i.e., Compass is the client), Compass will use the TLS secure communication mechanism to request the server's certificate when it makes a connection to the server. It will then authenticate the server and encrypt all communications with the server. Compass can optionally be configured to send the client's certificate when making a connection, which allows the server to authenticate the client as well.

When TLS is enabled on an **HL7 Source** (i.e., Compass is the server), Compass will send its configured certificate information to the client when it makes a connection to an encrypted listener. The client uses this information to authenticate and encrypt all communication with Compass. Compass can optionally be configured to require the client's certificate when making a connection to it. In this case, if the client does not send its certificate, the connection will be refused. The default is to allow missing client certificates (no client authentication), which is similar to how web browsers work.

On both the client and server sides, Compass may optionally be configured to accept self-signed certificates or to ignore certificate name mismatch errors. These options are primarily intended for testing purposes, and for optimal security, we recommend that these be left unchecked. TLS authentication certificates should ideally be obtained from a trustworthy TLS certificate authority (CA). If this is not feasible, a self-signed certificate can be generated for a local computer (see [Appendix E](#): for instructions), manually (and securely) copied to the remote computer, and installed into the certificate store on that computer as a trusted root certificate.

To properly configure incoming HL7 TLS connections, the [Listen Port](#) must be set on the [HL7 Sources pane](#), and the [TLS Certificate](#) path and certificate [Password](#) must be set on the [HL7 System pane](#). The [TLS Certificate](#) path should be set to the file system location of the certificate that Compass should present for identification to clients. It is suggested that the certificate be a standard PKCS#12 certificate, and it must contain an exportable private key. The [Password](#) must be set to the password for the private key in the certificate. Note: Using a certificate format that does not password-protect the private key allows the password setting to be ignored, but we highly recommend keeping private keys password-protected. This appendix does not describe the procedures that are required to obtain TLS certificates.

This configured certificate information will be provided to any HL7 TLS client that connects to Compass on a TLS listen port. This configured certificate information can also be provided to any server that Compass connects to, if configured to do so. All communication over this connection is then encrypted using the exchanged certificate(s).

2 Secure Communication with Compass Web

There are several configuration changes that can be made to the Windows computer on which Compass is installed to enhance the security of the Compass web interface. These changes assume that HTTPS (TLS) is used exclusively to access the Compass web interface (which we strongly recommend), and they primarily involve changes to the supported TLS protocols and ciphers. Some of these changes only apply to certain supported Windows versions, while others apply to all the supported Windows versions. These changes include the following:

- Disabling support for the SSL 3.0 protocol (if supported)
- Disabling support for the TLS 1.0 protocol (if older browsers must be supported)
- Enabling support for the TLS 1.1 and the TLS 1.2 protocols (if not already supported) and Disabling support for the TLS 1.1 protocol (for improved security)
- Disabling support for the RC4 cipher suite
- Disabling support for the Triple DES (3DES) cipher suite
- Enabling or Disabling support for the TLS 1.3 protocol (for improved security)

The changes discussed below can either be made manually (using `regedit`) or using scripts.

The Compass installation includes a set of Windows PowerShell scripts installed in the `C:\Program Files\Laurel Bridge Software\Compass\scripts` directory that can be used to perform these changes in an automated fashion. The scripts are all Authenticode-signed (by “Laurel Bridge Software, Inc.”) for security purposes, and they must run with the PowerShell execution policy set to allow signed scripts to run (such as `AllSigned`

or `RemoteSigned`). In addition, they must run from a PowerShell instance with Administrator privileges (needed to modify the registry).

The most portable way to start a PowerShell instance with Administrator privileges is to click on the Start button, type “powershell”, right-click on “Windows PowerShell” and select “Run as administrator” (note that you may need to answer the security prompt, if UAC is enabled). The PowerShell instance will show that it is running as Administrator in the title bar. Once it is running, enter the following command:

```
cd C:\Program Files\Laurel Bridge Software\Compass\scripts
```

To confirm that the execution policy will allow signed scripts to run, use the following command:

```
Get-ExecutionPolicy
```

If the execution policy returned is `Restricted`, then enter the following command:

```
Set-ExecutionPolicy AllSigned
```

The easiest way to confirm which protocols and ciphers are currently supported on a platform is to use the Nmap tool. Nmap is a network port scanner, available from <https://nmap.org>, that can optionally scan TLS ports and determine which protocols and ciphers are supported. After installing, use the following command to scan the HTTPS port:

```
nmap --script ssl-enum-ciphers -p <portnum> <hostname>
```

2.1 Disabling SSL 3.0 Support

SSL 3.0 is an older encryption protocol that is no longer considered cryptographically secure (due to its POODLE attack vulnerability). While the protocol is no longer supported by modern browsers, it is still supported by some older versions of Windows, including Windows 7 and Windows Server 2008 R2, so its use can still be negotiated by certain older browsers when connecting to these systems. We recommend that its use be disabled if running either of the above operating systems.

To disable SSL 3.0 manually, start Windows `regedit` and do the following:

- Navigate to:
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
- Add the subkey “SSL 3.0”.
- Under subkey “SSL 3.0”, add the subkeys “Client” and “Server”.
- Under subkey “Client”, add the DWORD entry “DisabledByDefault” and set it to the value “1”.
- Under subkey “Server”, add the DWORD entry “Enabled” and set it to the value “0”.

To disable SSL 3.0 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an `AllSigned` execution policy, as described above) and then run `DisableSSL3.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter `R` (= Run once) or `A` (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process. The change can be confirmed using `Nmap` to enumerate the supported protocols and ciphers on the HTTPS port, as described above.

2.2 Disabling TLS 1.0 Support

TLS 1.0 is an older encryption protocol that is considered cryptographically weak (due to its potential BEAST attack vulnerability when using obsolete browsers). We recommend that its use be disabled, unless all clients connecting to Compass will be using modern (evergreen) browsers such as Chrome, Firefox, and Edge.

To disable TLS 1.0 manually, start Windows `regedit` and do the following:

- Navigate to:
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
- Add the subkey "TLS 1.0".
- Under subkey "TLS 1.0", add the subkeys "Client" and "Server".
- Under subkey "Client", add the DWORD entry "DisabledByDefault" and set it to the value "1".
- Under subkey "Server", add the DWORD entry "Enabled" and set it to the value "0".

To disable TLS 1.0 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `DisableTLS10.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process. The change can be confirmed using `Nmap` to enumerate the supported protocols and ciphers on the HTTPS port, as described above.

2.3 Enabling or Disabling TLS 1.1 Support and Enabling TLS 1.2 Support

TLS 1.1 and 1.2 are newer encryption protocols that are considered cryptographically secure. TLS 1.2 has many security improvements over TLS 1.1. We strongly recommend using TLS 1.2 whenever possible and enabling TLS 1.1 only when it is necessary for communication with systems that are unable to support TLS 1.2.

Some older versions of Windows, including Windows 7 and Windows Server 2008 R2, do not support TLS 1.1 and 1.2 by default. If the Windows updates for the above operating systems are up-to-date (specifically, if KB3140245 is installed), the protocols themselves are available, but they must be manually enabled.

To enable TLS 1.1 and 1.2 manually, start Windows `regedit` and do the following:

- Navigate to:
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
- Add the subkey "TLS 1.1" and "TLS 1.2".
- Under subkey "TLS 1.1", add the subkeys "Client" and "Server".
- Under subkey "TLS 1.2", add the subkeys "Client" and "Server".

- Under subkey "Client" for both protocols, add the DWORD entry "DisabledByDefault" and set it to the value "0".
- Under subkey "Server" for both protocols, add the DWORD entry "Enabled" and set it to the value "1".

To enable TLS 1.1 and 1.2 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `EnableTLS11.ps1` and `EnableTLS12.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process. The changes can be confirmed using `Nmap` to enumerate the supported protocols and ciphers on the HTTPS port, as described above.

To disable the TLS 1.1 protocol using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `DisableTLS11.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

2.4 Disabling Support for the RC4 Cipher Suite

RC4 is a family of encryption ciphers that is no longer considered cryptographically secure (due to its NOMORE attack vulnerability – see RFC 7465 for details). Unfortunately, this cipher suite is supported by all three versions of TLS (1.0, 1.1, and 1.2), so its use can still be negotiated by certain modern browsers. We recommend that its use be disabled if possible (i.e., as long as all supported clients support other, more secure, ciphers).

To disable RC4 manually, start Windows `regedit` and do the following:

- Navigate to:
`HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers`
- Add the subkey "RC4 128/128".
- Add the subkey "RC4 40/128".
- Add the subkey "RC4 56/128".
- Under subkey "RC4 128/128", add the DWORD entry "Enabled" and set it to the value "0".
- Under subkey "RC4 40/128", add the DWORD entry "Enabled" and set it to the value "0".
- Under subkey "RC4 56/128", add the DWORD entry "Enabled" and set it to value "0".

To disable RC4 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `DisableRC4.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process. The change can be confirmed using `Nmap` to enumerate the supported protocols and ciphers on the HTTPS port, as described above.

2.5 Disabling Support for the Triple DES (3DES) Cipher Suite

Triple DES is a family of encryption ciphers that is no longer considered cryptographically secure (due to its SWEET32 attack vulnerability). Unfortunately, this cipher suite is supported by all three versions of TLS (1.0, 1.1, and 1.2), so its use can still be negotiated by certain modern browsers. We recommend that its use be disabled if possible (i.e., as long as all supported clients support other, more secure, ciphers).

To disable Triple DES manually, start Windows `regedit` and do the following:

- Navigate to:
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Ciphers
- Add the subkey "Triple DES 168".
- Under subkey "Triple DES 168", add the DWORD entry "Enabled" and set it to the value "0".

To disable Triple DES using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `Disable3DES.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process. The change can be confirmed using `Nmap` to enumerate the supported protocols and ciphers on the HTTPS port, as described above.

2.6 Enabling or Disabling TLS 1.3 Support

TLS 1.3 is an even newer protocol with speed and security enhancements. Compass supports this protocol. Currently, Microsoft includes an experimental version of TLS 1.3 in Windows 10, version 1909, which is disabled by default. Microsoft does not include TLS 1.3 in the Windows Server versions that Compass supports.

To enable this experimental version of TLS 1.3 on Windows 10, version 1909 manually, start Windows `regedit` and do the following:

- Navigate to:
HKLM\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
- Add the subkey "TLS 1.3".
- Under subkey "TLS 1.3", add the subkeys "Client" and "Server".
- Under subkey "Client" add the DWORD entry "DisabledByDefault" and set it to the value "0".
- Under subkey "Server" add the DWORD entry "Enabled" and set it to the value "1".

To enable TLS 1.3 on Windows 10, version 1909 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `EnableTLS13.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Once the above changes have been made (either manually or via PowerShell), restart Windows to complete the process.

Now on starting the Compass UI, a checkbox to enable the TLS 1.3 protocol is shown on the DICOM Destination Settings, DICOM Incoming Settings, HL7 Network Source Settings, HL7 Destination Settings, and Audit Logging Settings UIs.



This option is unchecked by default and may be checked to enable the TLS 1.3 protocol.

To disable the TLS 1.3 protocol on Windows 10, version 1909 using PowerShell, start a PowerShell instance (with Administrator privileges and at least an AllSigned execution policy, as described above) and then run `DisableTLS13.ps1` from the command line. If prompted whether or not to allow the LBS-signed script to run, enter R (= Run once) or A (= Always run).

Note that in addition the steps outlined above, additional steps may be necessary to enable support for TLS 1.3 on a per-browser basis for Chrome, Firefox, Edge, etc. Please refer to the documentation for these browsers.

3 A Note About FIPS 140-2 Compliance

Laurel Bridge Software neither provides nor configures the computer on which Compass runs and thus has no way to know a priori which cryptographic modules will be available in a particular installation. Since FIPS 140-2 compliance is based on the evaluation and certification of particular implementations of cryptographic modules (and not the algorithms themselves), Laurel Bridge Software cannot make any assertions regarding the FIPS 140-2 compliance of a specific Compass installation. Microsoft does provide a means by which a Windows system can be locked down so that only FIPS 140-2 compliant modules are used, but their current guidance no longer recommends running in this FIPS 140-2 compliant mode, unless doing so is explicitly required by government regulations. For more details on why this is so, see the article at <https://blogs.technet.microsoft.com/secguide/2014/04/07/why-were-not-recommending-fips-mode-anymore>.

Appendix D: Compass Configuration Backup Files

Compass' configuration is persisted in a file named `compass-config.xml`, and is typically located at:

```
"C:\ProgramData\Laurel Bridge Software\Compass\compass-config.xml".
```

Any time you save a new configuration by clicking “OK” on either the DICOM Options dialog or the HL7 Options dialog, a backup of your previous configuration is copied and saved to “`compass-config-autobackup-yyyy-MM-dd-HHmmss.xml`” in a subfolder called “backup”, where:

- `yyyy` is the year
- `MM` is the month
- `dd` is the day
- `HH` is the hour (in 24-hour format)
- `mm` is the minutes
- `ss` is the seconds

Compass will allow the user to configure whether or not the number of configuration backup files it keeps is limited (see the [System Settings](#) section for more information). If the user has chosen to limit the number of configuration backup files, then whenever a backup is created, Compass will only keep the configured number (keeping the most recent files). If the number of configuration backup files is not being limited and more than 20 backup files exist, a message box will be displayed reminding the user of how many backup files exist; with this setting, it is up to the user to remove any unwanted backup files.

Appendix E: Create and Export a Self-Signed TLS Certificate

1 Using IIS Manager

The following are instructions on how to create and export a self-signed certificate using Windows and Windows Server using the IIS Manager.

Once enabled, run the Internet Information Services (IIS) Manager program. Then perform the following actions:

1. Double-click "Server Certificates"
2. Under "Actions" (on the right side of the dialog), click "Create Self-Signed Certificate..."
3. A dialog will pop up; type in a friendly name (anything will do). The certificate store combo box should be "Personal". Click "OK".
4. Your certificate should now be in the list. Select the newly created certificate, and then click "Export..." under the "Actions" area (on the right side of the dialog).
5. Type in a filename and specify a password. Click "OK".
6. Verify that the file was created.
7. Import this certificate into Compass specifying the newly created file and corresponding password.

2 Using PowerShell

To create a self-signed TLS certificate using PowerShell, first start a PowerShell instance as Administrator. Then enter a command similar to one of the following (note that Tab-completion can be used to enter command, including parameter names):

```
PS> New-SelfSignedCertificate -CertStoreLocation Cert:\LocalMachine\My  
-DnsName "www.mydomain.com" # a normal TLS certificate
```

```
PS> New-SelfSignedCertificate -CertStoreLocation Cert:\LocalMachine\My  
-DnsName "*.mydomain.com" # a wildcard TLS certificate
```

```
PS> New-SelfSignedCertificate -CertStoreLocation Cert:\LocalMachine\My  
-DnsName "www.mydomain.com","ftp.mydomain.com" # a SAN TLS certificate
```

This will create the self-signed certificate and install it into the local machine repository. To export this certificate, run "certlm.msc", expand "Personal", click "Certificates", then right-click the newly-created certificate. Select "All Tasks", then "Export...". A wizard will pop up. Click Next to continue, choose to export the private key, then click Next, check the "Export all extended properties" under the .PFX section then click Next, enter a password (twice) then click Next, enter a filename for the .PFX file then click Next, and click Finish to generate the .PFX file containing the exported certificate (including the private key).

Appendix F: Hot Folder and HL7 Hot Folder Batch Behavior

Starting with version 2.8.1, Compass will have new capabilities for ingesting files via Hot Folder. This appendix explains some of the details of the new behavior, as well as how the configuration options affect that behavior.

1 Hot Folder Basics / Definitions

For both DICOM and HL7 Hot Folder Sources, the configuration has a directory called the **Path**, which is the folder location on the disk containing the files to be ingested into Compass. There is a configurable **Stability Time / Poll Time**, which is how long the files in the **Path** must have been present and unchanged in order to be ingested. Compass checks every **Stability Time** seconds to see if there are files which can be ingested.

For the purposes of this appendix, it is helpful to define several technical terms. First comes the technical term **ingest**. An **ingest** is simply a collection of files which Compass *considers* for import/addition to jobs. It is important to understand that an **ingest** is an ordered list of files; the details of that ordering are discussed more below.

The next term is a **batch**, which is a subset of an **ingest** (a **batch** might be all of the files in an **ingest**). For DICOM, Compass attempts to ingest a **batch** in a single "pseudo association", which gets an entry in the Association History table. This approach also affects how jobs are created within Compass, since Compass slots ingested instances into jobs on Association+StudyInstanceUid boundaries (assuming they all are going to one destination with Stable Studies disabled). So, for example, if during one **batch**, there are SOP Instances with three different Study Instance Uids in them, the **batch** would result in the creation of three different jobs in Compass. For HL7, Compass attempts to ingest a **batch** in a single "pseudo association", with each message constituting one job (again, assuming they are all going to one destination).

When files are ingested, any files which failed to ingest will either remain, or will be sent to the "jail" if the **Move non-ingested** is configured.

2 New Configuration Fields

Starting with Compass version 2.8.1, there are three new configuration items for a Hot Folder Source.

Max Files In Batch - This number specifies the maximum number of files Compass will ingest at a time. It will default to zero (which means "not in *batch mode*"), which is effectively the same as the old behavior (wait until the entire **Path** contents are stable before ingesting all of those contents).

Ingest Top Level Directories Separately - This checkbox indicates whether or not Compass will consider ingesting the entire contents of the **Path** as a single **ingest** (unchecked), or whether it

will consider each top-level directory in the path as a separate **ingest**, plus one more for the files (not contained in subdirectories) present in the **Path** (checked). Effectively, this allows the behavior to change from a single all-or-nothing **ingest**, to N+1 **ingests**, where N is the number of directories present at the top level of the **Path**. This will default to false, which is effectively the same as the old behavior.

Honor Poll Time Between Batches - This checkbox controls the timing between **batches**. When unchecked, Compass will proceed through the **ingests** from one **batch** to the next with no delay. When checked, Compass will delay the **Stability Time** after each **batch**, refreshing the **ingest** list after the delay (in case the **Path** contents have changed).

3 Hot Folder Behavior

With the above settings and definitions, the hot folder behavior is straightforward: there can be one or more **ingests**, and each ingest can have one or more **batches**, as described below:

1. Compass determines the list of **ingests**. It will either be one **ingest** (**Ingest Top Level Directories Separately** is false) for the entire **Path**, or it will be N+1 **ingests**, where N is the number of directories at the top level within the **Path**. The **ingest** list is ordered by directory name (alphabetical), and then the ingest for the "loose" files in **Path** (not contained in subdirectories) is the last **ingest**. Within each **ingest**, its own ordering follows the original ordering: listing all files/directories alphabetically, with a depth-first recursive search of any subdirectories encountered.

2. Compass determines whether it is in *batch mode* (**Max Files In Batch** > 0) or not.

3. Compass runs the **ingest** list and processes each **ingest**:

- * If Compass is not in *batch mode*, all of the files in the **ingest** must be stable in order to create/process a single **batch** (all of the files in that ingest).

- * If Compass is in *batch mode*, then it will grab a **batch** of stable files from the **ingest** and process them as a **batch**, repeating until all of the files of the **ingest** have been handled. Of course, it is possible that some of the files will have errors on import (e.g. file contents are malformed, no rule is matched, etc.). In *batch mode*, if the **Honor Poll Time Between Batches** is true, then Compass will only ingest the first **batch** in the list, delay by the **Stability Time**, and return to step 1 above after the delay.

4 DICOM Hot Folder Examples

Consider a HotFolderSource with Path = "rootdir" with the following file and directory contents (directories prefixed with **[D]**)

[D] rootdir:

b1, b2, b3, b50

[D] RandomDir

a1, a2, a3, a4, a5, , a100

[D] Subdir1

q1, q2, q3, q100

x1, x2, x3, x100

s1, s2, s3, s35

[D] Study1

1.dcm, 2.dcm, 3.dcm, 4.dcm, 5.dcm

[D] Study2

d1, d2, d3, d80

[D] Study3

c1, c2, c3, ... c10

z1, z2, z3, ... z75

This structure has 555 instances, some of which are "loose" in the rootdir (b1-b50, s1-s35, z1-z75), while the others are contained in subdirectories under the rootdir. How would this structure be processed by Compass?

Behavior Prior to Version 2.8.1 - All 555 files would need to be stable, and all 555 files would be processed in one pseudo-association. The file list for processing would be, in order, {b1...b50, a1...a100, q1...q100, x1...x100, s1...s35, 1.dcm... 5.dcm, d1... d80, c1...c10, z1...z75}. Because this is one association, the number of Store Job Records that would be created as a result would be however many different StudyInstanceUids were present in the successfully ingested SOP Instances contained in these various files (assuming they were all going to one destination).

Compass 2.8.1 and later, **Max Files In Batch** = 0, **Ingest Top Level Directories Separately** = false. As noted above, these are the defaults, so the behavior is exactly the same as the old behavior. For clarity, using the definitions created for this appendix, because **Ingest Top Level Directories Separately** is false, the **ingest** list contains only one **ingest**: the **ingest** for the rootdir, including its subdirectories. The file list for that **ingest** is the same as the old behavior: {b1...b50, a1...a100, q1...q100, x1...x100, s1...s35, 1.dcm... 5.dcm, d1... d80, c1...c10, z1...z75}. Because we are not in *batch mode* (**Max Files In Batch** = 0), all the files in this **ingest** must be stable for the **ingest** to be processed. It will be processed in one **batch** of 555 files.

Compass 2.8.1 and later, **Max Files In Batch** = 0, **Ingest Top Level Directories Separately** = true. Because **Ingest Top Level Directories Separately** is true, the **ingest** list contains one **ingest** for each directory in the rootdir, plus one **ingest** for the rootdir's files (excluding its directories). The file lists for these **ingests** are as follows:

Ingest 1 (for [D] RandomDir): {a1..a100, q1...q100, x1...x100}

Ingest 2 (for **[D]** Study1): {1.dcm, 2.dcm, 3.dcm, 4.dcm, 5.dcm}

Ingest 3 (for **[D]** Study2): {d1...d80}

Ingest 4 (for **[D]** Study3): {c1...c10}

Ingest 5 (for **[D]** rootdir, files only): {b1...b50, s1...s35, z1...z75}

Because we are not in *batch mode* (**Max Files In Batch** = 0), for each **ingest**, all the files in that **ingest** must be stable for the **ingest** to be processed. If some **ingests** are entirely stable while others are not, the stable **ingests** will be processed, and that processing will occur in one **batch** per **ingest** which includes all of the files in that **ingest**.

Compass 2.8.1 and later, **Max Files In Batch** = 49, **Ingest Top Level Directories Separately** = false. Because **Ingest Top Level Directories Separately** is false, the **ingest** list contains only one **ingest**: the **ingest** for the rootdir, including its subdirectories. The file list for that **ingest** is the same as the old behavior: {b1...b50, a1...a100, q1...q100, x1...x100, s1...s35, 1.dcm... 5.dcm, d1... d80, c1...c10, z1...z75}. Because we are in *batch mode* (**Max Files In Batch** > 0), the *stable* files in this **ingest** will be available to be batched (any that are unstable will be skipped). Assuming that all the files are stable for this example, then this **ingest** will result in the following **batches** (N.B. **batches** can, and will, cross directories when we are not ingesting the top level directories separately):

Batch 1: {b1... b49} **Batch 2**: {b50, a1, ... a48} **Batch 3**: {a49, a97}

Batch 4: {a98, a99, a100, q1, ... q46} **Batch 5**: {q47...q95} **Batch 6**: {q96...q100, x1, ... x44}

Batch 7: {x45, ... x93} **Batch 8**: {x94...x100, s1... s35, 1.dcm...5.dcm, d1, d2} **Batch 9**: {d3... d51}

Batch 10: {d52...d80,c1...c10, z1...z10} **Batch 11**: {z11...z59} **Batch 12**: {z60...z75}

Each **batch** is ingested on its own pseudo-association, and each **batch** can have failures and successes, resulting in 0 to 49 instances actually being ingested. The number of StoreJobRecords created by the **batch** will be equal to the number of different StudyInstanceUids present in the successfully ingested instances in the **batch** (again, assuming they all are going to one destination). Also note that if **Honor Poll Time Between Batches** is true, only the first **batch** in this list will be processed, and then there will be a delay; if the **Path** contents remain unchanged, this process will repeat through the **batch** list above.

Compass 2.8.1 and later, **Max Files In Batch** = 49, **Ingest Top Level Directories Separately** = true. Because **Ingest Top Level Directories Separately** is true, the **ingest** list contains one **ingest** for each directory in the rootdir, plus one **ingest** for the rootdir's files (excluding its directories). The file lists for these **ingests** are as follows:

Ingest 1 (for **[D]** RandomDir): {a1..a100, q1...q100, x1...x100}

Ingest 2 (for **[D]** Study1): {1.dcm, 2.dcm, 3.dcm, 4.dcm, 5.dcm}

Ingest 3 (for **[D]** Study2): {d1...d80}

Ingest 4 (for **[D]** Study3): {c1...c10}

Ingest 5 (for **[D]** rootdir, files only): {b1...b50, s1...s35, z1...z75}

Because we are in batch mode (**Max Files In Batch** > 0), we will run the **ingest** list, batching each one individually. As we do, the *stable* files in each **ingest** will be available to be batched (any that are unstable will be skipped). Assuming that all the files are stable for this example, then these **ingests** will result in the following **batches** (N.B. batches will not cross top-level directories when we are ingesting the top level directories separately):

Batch 1 (Ingest 1): {a1...a49} **Batch 2** (Ingest 1): {a50...a98} **Batch 3** (Ingest 1): {a99, a100, q1...q47}

Batch 4 (Ingest 1): {q48...q96} **Batch 5** (Ingest 1): {q97...q100, x1... x45} **Batch 6** (Ingest 1): {x46...x94}

Batch 7 (Ingest 1): {q95...q100}

Batch 8 (Ingest 2): {1.dcm, 2.dcm, 3.dcm, 4.dcm, 5.dcm}

Batch 9 (Ingest 3): {d1...d49} **Batch 10** (Ingest 3): {d50...d80}

Batch 11 (Ingest 4): {c1...c10}

Batch 12 (Ingest 5): {b1...b49} **Batch 13** (Ingest 5): {b50, s1...s35, z1...z13} **Batch 14** (Ingest 5): {z14...z62}

Batch 15 (Ingest 5): {z63...z75}

Again, each **batch** is ingested on its own pseudo-association, and each **batch** can have failures and successes, resulting in 0 to 49 instances actually being ingested. The number of **StoreJobRecords** created by the **batch** will be equal to the number of different **StudyInstanceUids** present in the successfully ingested instances in the **batch** (again, assuming they all are going to one destination). Also note that if **Honor Poll Time Between Batches** is true, only the first **batch** in this list will be processed, and then there will be a delay; if the **Path** contents remain unchanged, this process will repeat through the **batch** list above.

Appendix G: DICOMweb to Google Cloud Platform (GCP)

As of version 3.3.0 of Compass, users can route DICOMweb requests (STOW, QIDO, WADO) up to GCP via OAuth 2.0. This appendix is provided to serve as a springboard to get the user started and is not intended to be a replacement for the actual documentation hosted by Google. See the GCP documentation for more detailed configuration questions.

1 Google's Cloud Healthcare API

The subsequent screen captures shown below assume the following:

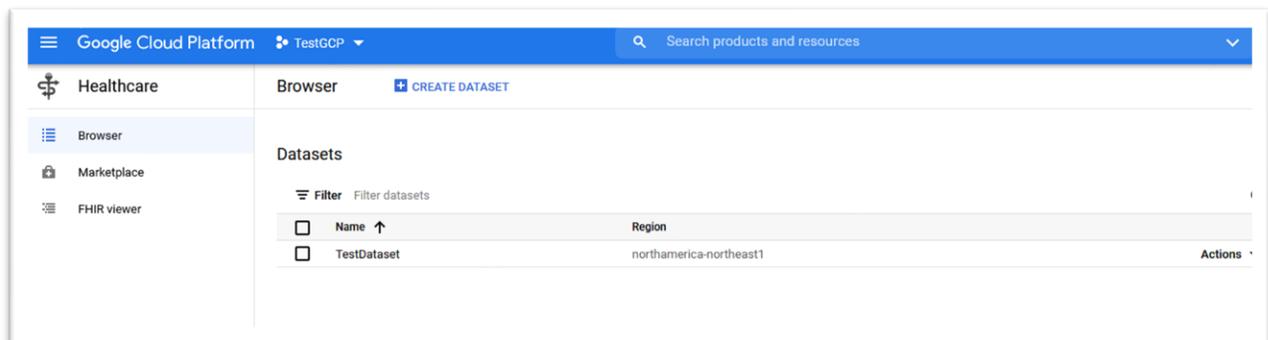
- A project has been created for use in the Google Cloud Platform
- The Cloud Healthcare API has been enabled for the aforementioned project
- Billing has been worked out for said Project and Healthcare API
- The appropriate users have already been created and granted the necessary permissions

Please see the online documentation provided by Google for help with any of the before mentioned prerequisites.

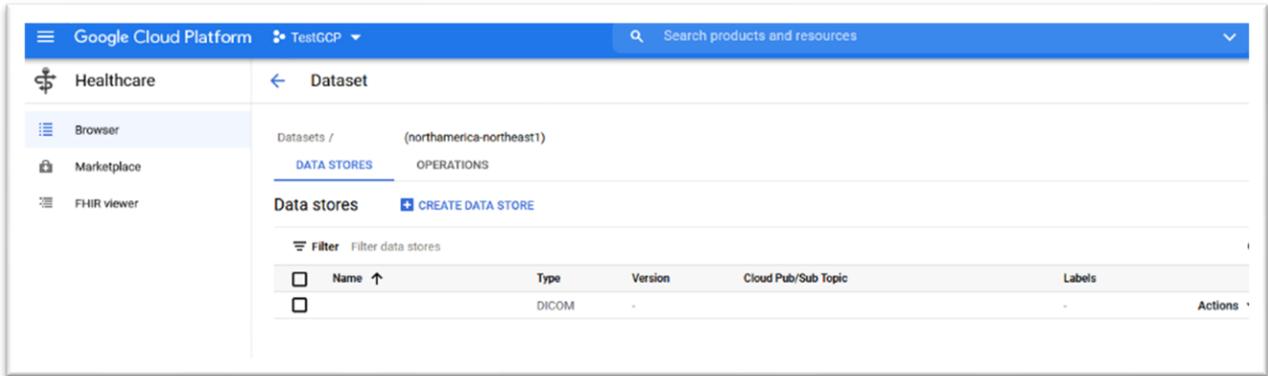
2 GCP Web Console

By this point, a project should have already been created and granted the necessary permissions for the Google Healthcare API. For the following screenshots, a sample project has been created, TestGCP, which will be used to demonstrate the various components of DICOMweb.

One of the first steps is to create a Dataset. (Please note that a Dataset in this context differs quite drastically from a Dataset as referred to by DICOM). The dataset requires the user specify a region where information uploaded to the cloud is to be stored for this dataset. The following screenshot demonstrates a dataset named TestDataset, located in the region northamerica-northeast1.

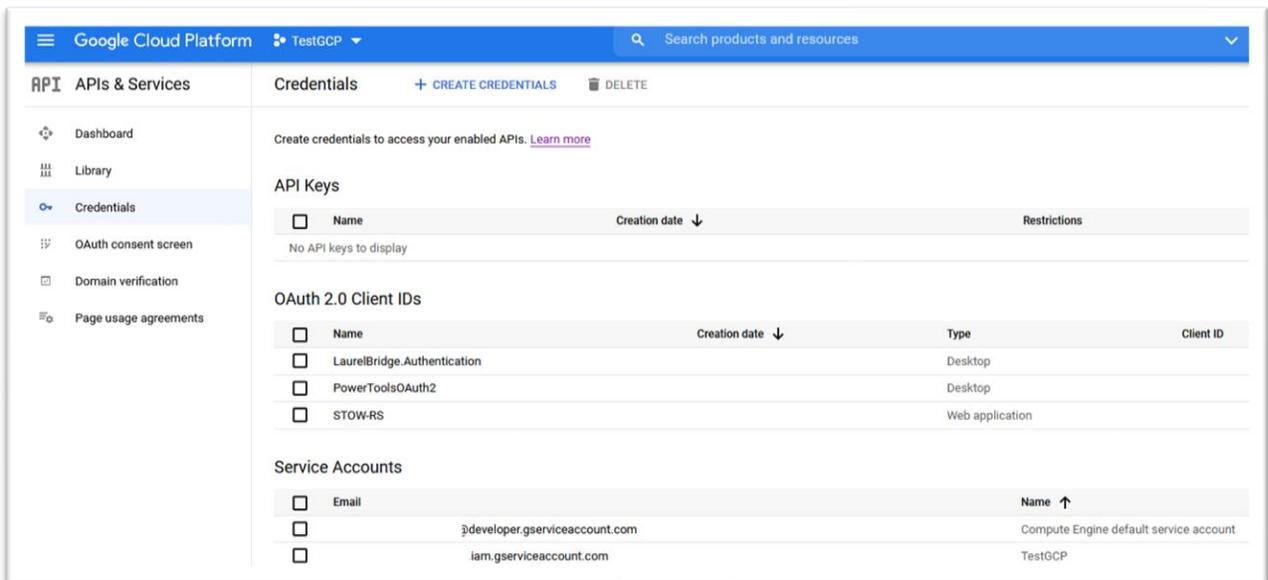


Once the dataset has been created, the next step is to create a data store. Datasets can contain multiple data stores. For the purpose of this short tutorial, a test data store has been created, of the type DICOM. Keep in mind Google supports multiple data store types, which is outlined in greater detail online in Google’s help documentation.



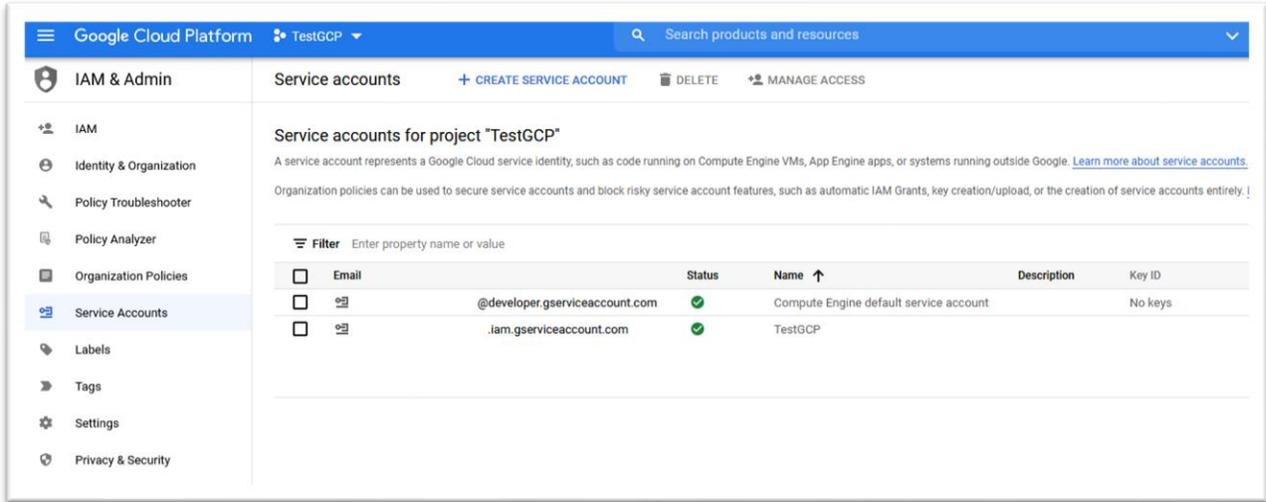
GCP does not accept requests from anonymous applications. As such, application credentials supported by GCP are API Keys, OAuth 2.0 Client IDs, and Service Accounts. The screen capture below has multiple OAuth 2.0 client IDs and Service Account keys for the TestGCP project.

Though OAuth 2.0 can be used to authenticate (and authorize) both user and service accounts, Compass requires the use of a service account, given the lack of a user directly tied to the Compass Service. That is another way of saying a user is not available to handle browser redirects required to authorize a user with the OAuth server.



The above screen capture shows a service account that was created for the TestGCP project with the service account id ending with iam.gserviceaccount.com. This is the service account id Compass will refer to in order to authorize communication with the Google Cloud Platform.

The next screen outlines the list of service accounts present for the TestGCP project. As shown below, the service account ending in iam.gserviceaccount.com has been created and added to the TestGCP project. This table is also where the user goes to manage Keys for each service account. (Note the keys have been blanked out but would otherwise be visible in the table).



Under the Actions Column (not visible in the above screen capture), the option to ‘Manage Keys’ is available. This screen allows the user to Add or Delete both P12 Certificate keys and JSON keys.

2.1 Authentication Method for GCP Service Account

GCP provides the option to download a generated JSON key file containing all the information necessary to authenticate with GCP for the service account. Again, this option can be found from the IAM & Admin section of the GCP web console, under the Service Accounts tab for a selected service account.

Also available is the option to upload an existing P12 certificate, which can be used to authenticate the service account.

In either scenario, keep the information gathered in a safe location until the Compass service can be properly configured for the service account mentioned above.

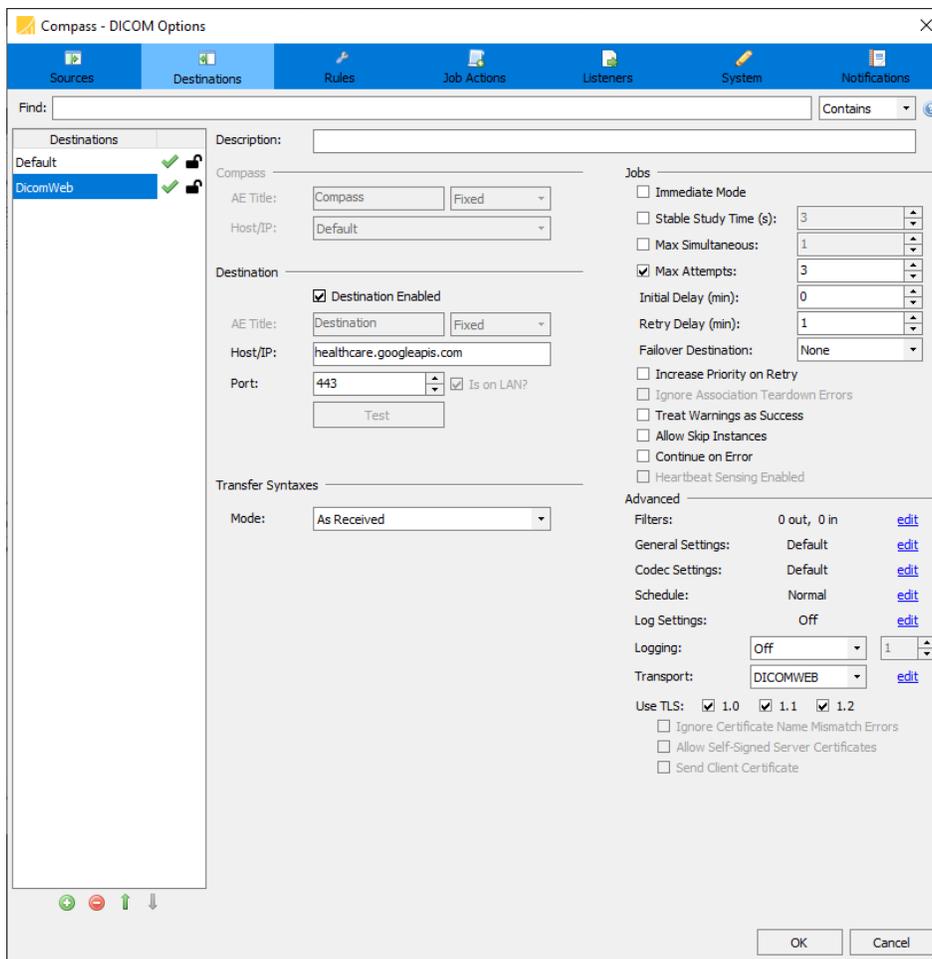
3 DICOMweb Destination

Once GCP has been properly configured for service account authorization and authentication, we can begin setting up Compass for DICOMweb support.

The first step is to create a new destination, in this case named 'DicomWeb'. We will be using this destination to forward DICOMweb requests directly to GCP over HTTPS, so the Host/IP is set to healthcare.googleapis.com, and the port to 443 (for HTTPS). It's important that the TLS checkboxes be enabled here to the request gets sent over HTTPS and not HTTP.

The other critical piece for this destination is that the Transport drop down gets set to DICOMWEB instead of DICOM.

The other configuration items may be adjusted as needed and desired, depending on the situation.



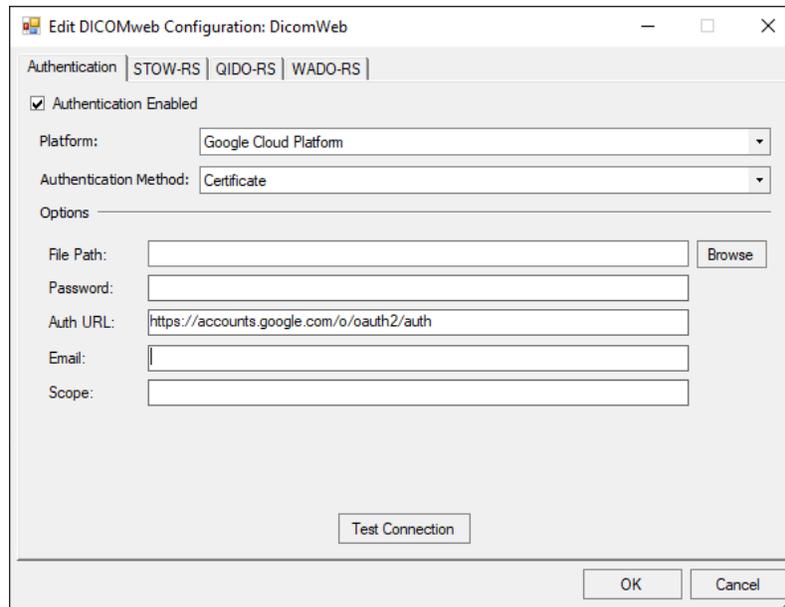
3.1 DICOMweb Configuration

Once the Compass destination for DICOMweb has been created, the ‘edit’ link can be selected to bring up the configuration form for that destination.

3.1.1 Authorization

The first section presented to the user is the Authentication tab, where we will set up GCP using whichever authentication method was selected in the GCP web console for the service account.

3.1.1.1 P12 Certificate

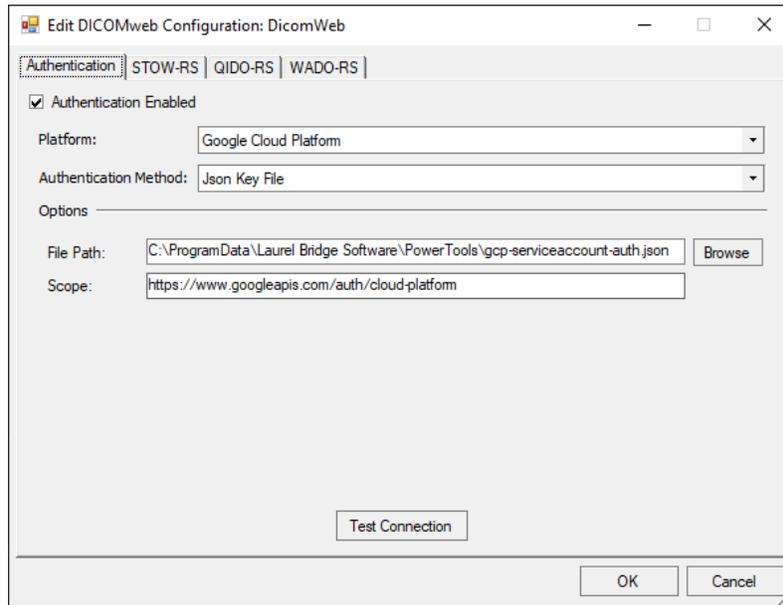


As shown in the above screen capture, the following options must be provided when selecting the Certificate Authentication Method:

- File Path – The full path on disk where the certificate is located. Note this must be a P12 certificate used for verification and encryption purposes and can be in PFX format.
- Password – The password for the private p7b certificate.
- Auth URL – The authorization URL for GCP. This URL is not always necessary and is auto populated by Compass. Adjust, as necessary.
- Email – The email address of the service account in GCP where this certificate has been uploaded.
- Scope – The comma-separated list of scopes to request authentication in GCP.

Once all the information has been populated, the user can select ‘Test Connection’ to submit the access token request to GCP. If successful, a bearer (access) token is printed out in the pop-up dialog to signify a successful authentication (and authorization) request to the GCP OAuth server.

3.1.1.2 Json Key File



As shown above, the following information must be specified when selecting the Json Key File Authentication Method:

- File Path – The full path on disk of the JSON key file downloaded from GCP when setting up a service account for a given project. This JSON key file contains all the critical pieces necessary to authentication (and authorize) the service account named in the key file.
- Scope - The comma-separated list of scopes to request authentication in GCP.

Once all the information has been populated, the user can select ‘Test Connection’ to submit the access token request to GCP. If successful, a bearer (access) token is printed out in the pop-up dialog to signify a successful authentication (and authorization) request to the GCP OAuth server.

3.1.2 STOW-RS, QIDO-RS, and WADO-RS with GCP

The remaining tabs located in the DICOMweb configuration form (not shown) contain the pieces necessary to perform a DICOMweb operation with GCP. At the time of this writing the following information is true:

- The ‘Resource Path’ must contain the full path to GCP given the following format:
 - [version]/projects/[project_id]/locations/[location]/datasets/[dataset]/dicomStores/[dicomstore]/dicomWeb/studies
 - version – The version of the GCP API to communicate with, such as v1
 - project_id – The id of the project created in GCP Healthcare API
 - location – The location selected for the aforementioned project.
 - dataset – The dataset created in the GCP Healthcare API for the aforementioned project.
 - dicomstore – The name of the dicom store, within the named dataset, for the identified project.

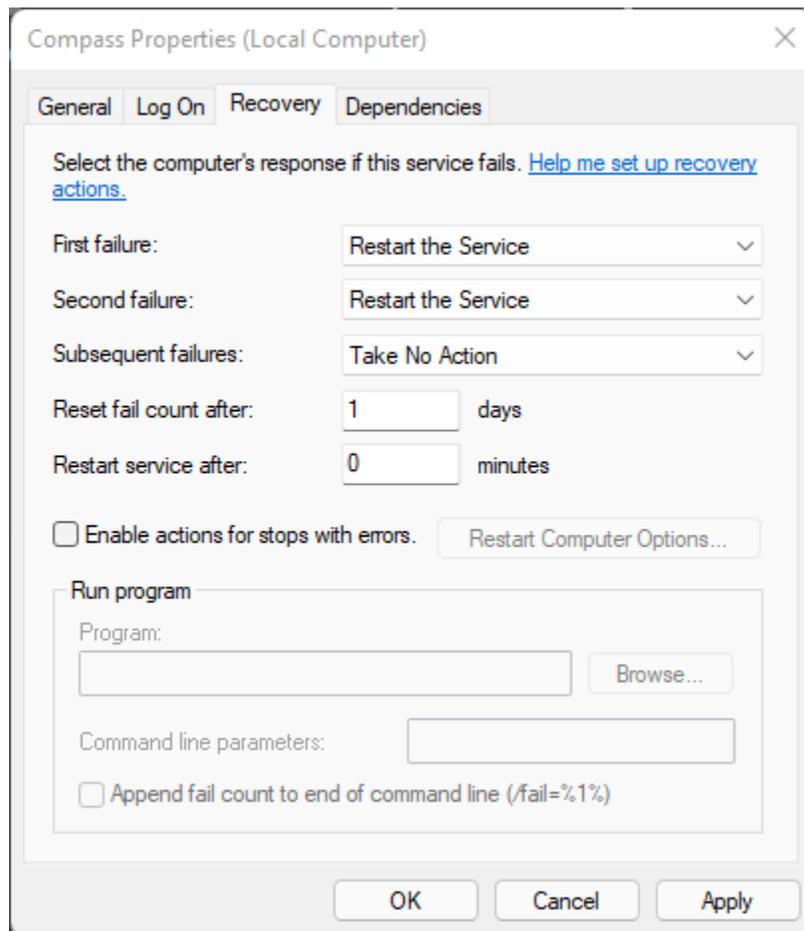
- The 'Response Media Type' must match exactly what GCP supports for the given protocol. For instance, we have had issues sending a QIDO request to GCP using a Response Media Type of anything other than 'application/dicom+json'.

Appendix H: Compass Service Crash: Restart Behavior and Logging

1 Restart Behavior

The Compass service is configured to restart automatically and without delay after the first and second failure. Subsequent failures do not result in an automatic restart. Failure counts are reset after one day.

These settings can be changed via the “Recovery” tab on the Properties dialog for the Compass service.



If the Compass service is part of a Windows Failover Cluster, the default Recovery settings may need to be changed.

2 Logging

If the Compass service crashes, a file called a “crash dump” will automatically be created in order to aid in troubleshooting the issue. The file will be created in the folder **C:\ProgramData\Laurel Bridge Software\Compass**. The crash dump files will have an extension of

.dmp, and at most three crash dump files will be created (so as not to consume too much disk space) before older crash dump files are overwritten.

These settings can be changed via the Windows Registry key:

[Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps\CompassService.exe](#)