Laurel Bridge PowerTools
User Manual

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

Document Version: 3.5.0
Document Number: LBDC-000062-030500
Last Saved: 11/15/2022 10:10:00 AM
Contents
1. PowerTools Introduction .................................................................................................. 1
   1.1. Overview .................................................................................................................. 1
   1.2. Included Utilities .................................................................................................... 1
       1.2.1. Graphical User Interface (GUI) Applications ................................................. 1
       1.2.2. Command Line Applications ........................................................................... 2
       1.2.3. Special Purpose Applications ........................................................................... 2
   1.3. Important Features .................................................................................................. 3
2. Installation ....................................................................................................................... 4
   2.1. Prerequisites ........................................................................................................... 4
   2.2. Installing PowerTools .............................................................................................. 4
       2.2.1. Version Rollback .............................................................................................. 4
       2.2.2. Problem with uninstalling ................................................................................. 4
3. Getting Started ............................................................................................................... 5
   3.1. Installing a License .................................................................................................. 5
   3.2. Activating a License over the Internet ..................................................................... 5
   3.3. Problem with Application Error on License Activation ........................................... 6
   3.4. PowerTools Launcher ............................................................................................... 7
   3.5. PowerTools Console ................................................................................................ 8
   3.6. Security .................................................................................................................. 9
       3.6.1. Steps to enable or disable FIPS enforcement for all PowerTools application ...... 9
       3.6.2. Steps to enable or disable FIPS enforcement for an individual application ...... 9
4. GUI Applications ........................................................................................................... 10
   4.1. Common Components .............................................................................................. 10
       4.1.1. Menu Bar .......................................................................................................... 10
       4.1.2. Options Dialog ................................................................................................ 11
       4.1.3. Filter Editor Dialog .......................................................................................... 11
       4.1.4. Client Routing Information .............................................................................. 16
       4.1.5. Server Routing Information ............................................................................ 17
       4.1.6. Transfer Syntaxes for Presentation Context Negotiation ................................. 17
       4.1.7. Application Window Renaming ....................................................................... 18
       4.1.8. External Data Dictionary Management ............................................................. 19
4.2. DICOM Client Applications ............................................................................................................. 20
  4.2.1. Query Client ................................................................................................................................. 20
  4.2.2. Store Client ................................................................................................................................. 26
  4.2.3. Study Generator ......................................................................................................................... 27
  4.2.4. Verification Client ..................................................................................................................... 33
4.3. DICOM Server Applications .......................................................................................................... 34
  4.3.1. Store Server ............................................................................................................................... 34
  4.3.2. Verification Server .................................................................................................................... 39
4.4. DICOM Data and File Manipulation Applications .............................................................................. 40
  4.4.1. DICOMDIR Creator .................................................................................................................. 40
  4.4.2. DICOM File Editor .................................................................................................................... 41
  4.4.3. DICOM Filter ............................................................................................................................. 54
  4.4.4. Dictionary Finder ....................................................................................................................... 54
  4.4.5. PDF to DICOM ........................................................................................................................... 56
  4.4.6. Structured Report Extractor ....................................................................................................... 56
4.5. HL7 Applications ............................................................................................................................. 61
  4.5.1. HL7 Receiver ............................................................................................................................. 61
  4.5.2. HL7 Sender ................................................................................................................................. 61
5. Advanced Filtering .............................................................................................................................. 66
  5.1. Composer Filter Action .................................................................................................................. 66
  5.2. De-Identify Filter Action .............................................................................................................. 68
    5.2.1. DICOM Attribute Tags to Anonymize ..................................................................................... 69
    5.2.2. General Settings ..................................................................................................................... 70
    5.2.3. Pixel Regions to Anonymize ................................................................................................. 71
    5.2.4. One-Way Anonymization ....................................................................................................... 74
    5.2.5. Modify Dates and Times ......................................................................................................... 75
    5.2.6. Additional Notes ..................................................................................................................... 78
  5.3. Re-Identify Filter Action .............................................................................................................. 78
      5.3.1. Additional Notes ..................................................................................................................... 79
6. Console Applications .......................................................................................................................... 80
  6.1. Overview .................................................................................................................................... 80
  6.2. DICOM to PDF ............................................................................................................................. 81
  6.3. Indexer ....................................................................................................................................... 81
6.4. Modality Simulator .................................................................................................................................. 81
6.5. MWL Client........................................................................................................................................... 81
6.6. MWL Server ........................................................................................................................................... 82
6.7. QIDO Client ........................................................................................................................................... 82
6.8. QIDO Server ........................................................................................................................................ 83
6.9. Query Client (console version) ............................................................................................................. 84
6.9.1. Query example ................................................................................................................................. 84
6.9.2. Creating a Filter File ...................................................................................................................... 85
6.9.3. Format of Query Configuration File .............................................................................................. 86
6.10. Query Server ...................................................................................................................................... 87
6.11. RIS Simulator ..................................................................................................................................... 87
6.12. STOW Client ...................................................................................................................................... 88
6.13. STOW Server .................................................................................................................................... 89
6.14. WADO Client ..................................................................................................................................... 89
6.15. WADO Server ................................................................................................................................... 89
7. Overriding DICOM Framework Defaults ................................................................................................. 90
8. Application Summary ............................................................................................................................... 91
8.1. Combined GUI and Console Applications ......................................................................................... 91
8.2. GUI-only Applications ....................................................................................................................... 91
8.3. Console-only Applications ................................................................................................................ 91
9. PowerTools Quick Reference .................................................................................................................. 92
10. Conductor ............................................................................................................................................... 94
10.1. Overview of Conductor .................................................................................................................... 94
10.2. Installing Conductor ........................................................................................................................... 95
10.2.1. Single System ............................................................................................................................... 95
10.2.2. Virtual Network .......................................................................................................................... 95
10.3. Getting Started with Conductor ........................................................................................................ 96
10.3.1. Installing a License ....................................................................................................................... 96
10.3.2. Starting the Conductor Agent Service ......................................................................................... 96
10.3.3. Starting the Conductor Controller ............................................................................................. 97
10.3.4. Conductor User Interface ......................................................................................................... 97
10.4. Conductor Scenarios .......................................................................................................................... 108
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4.1.</td>
<td>What is a scenario?</td>
<td>108</td>
</tr>
<tr>
<td>10.4.2.</td>
<td>Configuring a scenario</td>
<td>108</td>
</tr>
<tr>
<td>10.4.3.</td>
<td>Running a scenario from the user interface</td>
<td>109</td>
</tr>
<tr>
<td>10.4.4.</td>
<td>Running a scenario from the command line</td>
<td>113</td>
</tr>
<tr>
<td>10.5.</td>
<td>Conductor Troubleshooting</td>
<td>114</td>
</tr>
</tbody>
</table>
1. PowerTools Introduction

1.1. Overview

Laurel Bridge PowerTools™ is a software only collection of stand-alone Windows® applications designed for the diagnosis, testing, and/or repair of DICOM® communications, for the viewing, repair, or creation of DICOM data sets and their contents, and for communications testing of basic HL7 messaging. It is suitable for use by PACS administrators, integration engineers, system testers, software developers, and others who have need of a broad range of DICOM related tools and utilities or need to test HL7 messaging.

The stand-alone PowerTools programs operate as easy-to-use DICOM server (SCP) or client (SCU) applications for communication purposes or operate as other types of applications capable of viewing, changing, creating, converting, or fixing a DICOM data set, or sending and receiving of HL7 messages. Many are provided as both GUI-based and command line-based applications, so the PowerTools utilities may be used as independent applications or scripted into a variety of situations where their capabilities are needed.

These tools and utilities may be used to transmit, receive, or edit DICOM medical data or to send and receive HL7 messages. However, the PowerTools applications are not intended for diagnostic review of images or other medical data.

1.2. Included Utilities

The core utilities officially included in the PowerTools suite are:

1.2.1. Graphical User Interface (GUI) Applications

These are PowerTools applications that have a graphical user interface. Many also have command line versions.

- **PowerTools Launcher** – provides access to all PowerTools GUI applications.
- **DICOMDIR Creator** – creates a DICOMDIR file from a directory of DICOM images on disk (i.e. DICOM media creator).
- **DICOM File Editor** – allows the viewing, printing, and modification of a DICOM file on disk, including modification of sequence elements. Pixel data can also be viewed.
- **DICOM Filter** – filters one or more DICOM images on disk.
- **Dictionary Finder** – displays data from the DICOM data dictionary.
- **HL7 Receiver** – receives HL7 messages from an HL7 client or sender.
- **HL7 Sender** – sends HL7 messages to an HL7 server or receiver.
- **PDF to DICOM** – create DICOM Encapsulated PDFs from existing PDF files and write them to files or store them to a PACS.
- **Query Client** – queries a query SCP (or PACS) for DICOM images and can move DICOM images from the server to another location.
- Store Client – stores DICOM images from disk to a store SCP.
- Store Server – receives C-Store DICOM images from a store SCU.
- Structured Report Extractor – parses structured reports and creates mapping files for use with Laurel Bridge Software Compass™ and third-party systems.
- Study Generator – generates DICOM images and stores them to a store SCP.
- Verification Client – verification SCU to send C-Echo requests.
- Verification Server – verification SCP to receive C-Echo.

1.2.2. Command Line Applications

These console-based PowerTools applications include command line versions of some of the GUI-based applications listed above and a number that are console-only.

- DICOMDIR Creator – creates a DICOMDIR file from a directory of DICOM images on disk (i.e. DICOM media creator).
- DICOM File Editor – dumps the contents of a DICOM file and edits a single DICOM file.
- DICOM Filter – filters one or more DICOM images on disk.
- DICOM to PDF – extracts encapsulated PDFs from DICOM files.
- Dictionary Finder – displays data from the DICOM data dictionary.
- Indexer – iterates over images and indexes specified tags into a csv file.
- MWL Client – connects to a modality worklist SCP.
- MWL Server – provides a modality worklist SCP for a directory on disk.
- QIDO Client – provides a DICOMweb QIDO-RS client.
- QIDO Server – provides a DICOMweb QIDO-RS server.
- Query Client – console version, queries a query SCP (or PACS) for DICOM images and can move DICOM images from the server to another location. Demonstrates the use of command line config and filter files.
- Query Server – provides a query SCP for a directory on disk.
- RIS Simulator – sends HL7 ORM messages to a Modality Worklist Server.
- Study Generator – generates DICOM image studies and stores or writes them.
- Verification Client – verification SCU to send C-Echo requests.
- Verification Server – verification SCP to receive C-Echo.
- WADO Client – provides a DICOMweb WADO-RS client.
- WADO Server – provides a DICOMweb WADO-RS server.

1.2.3. Special Purpose Applications

- Conductor – an additional-cost application controller and performance measurement tool, used for performance testing proposed scenarios on a virtual network as a means of proof-of-concept or predictive analysis of a network configuration. See the Conductor section that begins below on page 94.
Note that occasionally some new tools which are not listed above may be included in intermediate release versions of PowerTools, providing an early look at upcoming additions to the suite.

1.3. Important Features

All applications in PowerTools have been written with easy to understand interfaces. To make PowerTools even easier to use, each application has built-in features designed to make understanding and using the application a more natural and less frustrating experience.

In all GUI applications many items on the screen have a tooltip which can be displayed by hovering the mouse pointer over the item. This tooltip displays information about the item which can be helpful for understanding the item’s purpose.

For console applications, running the applications with the -h help flag will display the usage for the application, listing a description of all possible flags and arguments. Using the -h help flag with GUI applications will display usage information in a pop-up window, while adding the -nogui flag to many of the GUI applications will display command-line usage for the console versions of the applications.

Many of the GUI and command line software applications in PowerTools incorporate innovations in user interface ease-of-use and the robustness of the underlying DICOM framework or HL7 library upon which most of the Laurel Bridge PowerTools applications are built.

An interesting innovation that is built into the Study Generator command line tool is the inclusion of a first and last name database culled from recent U.S. Census Bureau statistics, enabling the random generation of realistic-looking patient names in roughly the same frequency with which the names occur in the general population.
2. **Installation**

To install PowerTools run the installer executable (Setup.exe) and follow the on-screen directions.

2.1. **Prerequisites**

For PowerTools to function correctly, the following must first be installed or present on the target system:

- Microsoft .NET Framework 4.8, minimum
- Microsoft ASP.NET Core Runtime 3.1 Hosting Bundle

The prerequisites should be installed **prior to** running the PowerTools.msi installer. This step might have already been performed on a fully updated system, and otherwise is generally required only when installing a version of PowerTools on a system for the first time.

2.2. **Installing PowerTools**

After installing the prerequisites, run the PowerTools.msi installer. For machines with an older version of PowerTools installed, this installer will upgrade a previous installation while maintaining any current configuration settings. In the rare circumstance where installing a newer version over an older version of PowerTools fails, uninstalling the older version first typically resolves the problem.

2.2.1. **Version Rollback**

If it is desired to install an older version of PowerTools than the currently installed version, or to reinstall the current version, it is necessary to first uninstall the current version.

2.2.2. **Problem with uninstalling**

When a newer release of PowerTools is installed, any previous version is uninstalled. If the **Conductor** service is not present (e.g., an admin user previously ran `sc delete Conductor`) the uninstallation of PowerTools may fail with a “Problem uninstalling Conductor service” error. This is resolved by running the following commands in a console with Administrator privileges:

**PowerTools versions 1.0.0 to 1.0.46**

```bash
set PTPATH=C:\Program Files (x86)\Laurel Bridge Software\PowerTools
sc create Conductor binPath= "%PTPATH%\Conductor.exe"
```

**PowerTools versions 2.0.0 and newer**

```bash
set PTPATH=C:\Program Files\Laurel Bridge Software\PowerTools
sc create Conductor binPath= "%PTPATH%\Conductor.exe"
```
3. **Getting Started**

3.1. **Installing a License**

The first step after launching PowerTools is to install a license. Running the PowerTools Launcher application from the start menu (see below) for the first time will bring up a dialog allowing a license to be installed. When the About Laurel Bridge PowerTools dialog appears, click the Install License… button, browse to the license file, and hit OK. 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... in any PowerTools GUI application. The license works on a per user basis; each user will need to install a valid license to run any of the utilities.

![PowerTools User Interface](image)

3.2. **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 (Product S/N) to activate your license. All fields on the form should be filled out.

When the License Activation dialog first appears, it will attempt to contact the Laurel Bridge license server and pre-populate the form with as much information as it has available. If PowerTools cannot reach the license server you will be presented with an information message describing which fields need to be entered to activate your license on the Manual Product Activation interface on the Laurel Bridge Customer Access web page here:

Note that if PowerTools does not have internet access you will need the 16-digit Activation Request Code (ARC) from the Manual Product Activation (see above). On that form enter your 16-digit Product Serial Number and the 16-digit Activation Request Code. After completing the steps on the web activation form you can download a fully activated PowerTools license. Once downloaded, use the Install License... button on the About dialog to install your activated license.

If you only have a MAC address you will need a valid Customer Login account for laurelbridge.com to proceed. For assistance with this process, please send email to: support@laurelbridge.com

### 3.3. Problem with Application Error on License Activation

If you are presented with the License Activation Dialog after installing or upgrading and you click the Activate button and PowerTools immediately crashes, you may be having an issue with the Microsoft Visual C++ Redistributable for Visual Studio 2017 x64 version, which is normally installed as part of the standard PowerTools installation. If you see this error, installing all available Windows system updates should fix the problem. To specifically install the Redistributable package only, search online for the most recent “vcredist x64” found on the Microsoft.com web site.
3.4. PowerTools Launcher

The PowerTools Launcher can be found in the Start menu and provides access to all PowerTools GUI applications. Each application has a brief description available by selecting the application in the list. Double-clicking an application or selecting it in the list and clicking **Launch** will begin execution of that application. All PowerTools applications (except for the Launcher) can have multiple copies running at once. Information about PowerTools and access to license information can be accessed by clicking the **About PowerTools** button. The user manual you are reading is available via the **User Manual** button.

The DICOM logo in the upper left brings up the external data dictionary manager and indicates what dictionary is in use. Options are the default dictionary, the built-in DICONDE dictionary, or a custom, user-designed dictionary, as described in section 4.1.8.
3.5. PowerTools Console

The **PowerTools Console** can be found in the **PowerTools Launcher** and the start menu. It provides access to all PowerTools command-line (console) applications. The console will launch and set the working directory to the installation folder for PowerTools. It will also add the PowerTools installation folder to the Path temporarily for the session.

Running the predefined PowerTools console command **list** will display all executables as well as any example configuration files and documentation files (see below). Running the command **dir/s *.exe** will display all executables in the directory and subdirectories. Running any of the applications with the **-h** flag will display the command-line usage statement for that application.

Many of the applications are combined GUI and console applications. For these combined applications, the **-h** flag will display usage information for the GUI version, while using the **-h** and **-nogui** flags together will display usage information for the console version. For GUI-only and console-only applications, the **-h** flag is all that is needed to display usage information.

![Example output produced by running the PowerTools console command list.](image)
3.6. Security

Some systems are configured using the Federal Information Processing Standard (FIPS) for executable application security. As installed, PowerTools applications have this feature disabled. A simplified work-around is provided to enable and disable FIPS enforcement on an all-application or an application-by-application basis.

3.6.1. Steps to enable or disable FIPS enforcement for all PowerTools application

- Navigate to the PowerTools installation directory (C:\Program Files\Laurel Bridge Software\PowerTools) or run the PowerTools Console from the Launcher application if possible.
- Inside that directory run the fips.bat script with the following usage:

```
Usage: fips.bat enable|disable [file]
  enable - sets enforceFIPSPolicy to true for all exe.configs
  disable - sets enforceFIPSPolicy to false for all exe.configs
  file - optionally, indicate a file (or pattern ending with .exe.config)
```

- Examples:
  - fips enable
  - fips disable Store*.exe.config
  - fips enable blah

3.6.2. Steps to enable or disable FIPS enforcement for an individual application

- Navigate to the PowerTools installation directory (C:\Program Files\Laurel Bridge Software\PowerTools) or run the PowerTools Console from the Launcher application if possible.
- Identify the PowerTools application or applications you would like to execute, noting their filenames. For example: DicomFileEditor.exe or StoreServer.exe
- For each of those applications, identify their corresponding exe.config files: DicomFileEditor.exe.config or StoreServer.exe.config
- Using a text editor, edit each of the identified exe.config files, changing the enabled value of the enforceFIPSPolicy to true (enable FIPS enforcement policy) or false (disable FIPS enforcement policy). The config file looks like this (enable value is bolded):

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.8"/>
  </startup>
  <runtime>
    <gcAllowVeryLargeObjects enabled="true"/>
    <enforceFIPSPolicy enabled="false" />
    <AppContextSwitchOverrides value="Switch.UseLegacyAccessibilityFeatures=false;Switch.UseLegacyAccessibilityFeatures.2=true"/>
  </runtime>
</configuration>
```

- Once the exe.config files have been edited, you should be able to successfully run the corresponding applications on an otherwise FIPS enabled system.
4. GUI Applications

4.1. Common Components

There are several components that are common to many or all PowerTools GUI applications. Learning to use these components will make interacting with the PowerTools GUI interface much easier.

4.1.1. Menu Bar

The Menu Bar may vary slightly between GUI applications, although the following menu items and commands are present in almost every PowerTools application.

- **File > Exit** – Exits the application.
- **Tools > Options** – Displays options for the application. Many applications have settings to specify:
  - Logging
  - Filtering
- **Tools > Revert To Factory Settings** – Reverts all configuration settings for the application back to their original defaults.
- **Help > Support Request** – Opens the Send Support Request dialog so a user can send a request for support directly from the running application.
- **Help > Feedback** – Opens the Send Feedback dialog so a user can send feedback to Laurel Bridge Software directly from the running application.
- **Help > About** – Shows the About dialog for PowerTools information and for license management, including support for license installation and activation.
## 4.1.2. Options Dialog

The Options Dialog provides two common features shared by most of the GUI applications: logging and filtering.

- **Logging**
  - **Enable Logging** – enables/disables logging for the application along with enabling/disabling pruning of older or overly plentiful log files.
  - **Log Level** – selects the amount/verbosity of the logging (useful for debugging).
  - **View Log** – click to open the current log for the application.
  - **View Log Directory** – click to open the directory where the logs are stored.

- **Filtering** (only present in the Options Dialog for client and server applications)
  - **Enable Filtering** – enables/disables filtering for the application’s input/output DICOM messages.
  - **Edit Filters…** – opens the Filter Editor Dialog to edit the filters for the application.

- **OK** – exits the Options Dialog and saves any changes.
- **Cancel** – exits the Options Dialog without saving.

## 4.1.3. Filter Editor Dialog

The Filter Editor Dialog provides the ability to create filters which will modify DICOM communication or DICOM files on disk. Each filter is chained together to the next one and all are executed in order on a DICOM data set—a collection of DICOM elements. These elements comprise the header of each DICOM image filtered (currently pixel data cannot be filtered).
Key components in the Filter Editor Dialog include:

- **Type** – changes the filters that are being edited (for example, “Inbound Filters” in the image above represents all incoming DICOM communication to the application).
- **Import…** – imports a filter configuration from disk (allows configurations to be shared between PowerTools utilities).
- **Export…** – exports a filter configuration to disk (allows configurations to be shared between PowerTools utilities).
- **Filter** – the list of filters in order of execution (filters are chained together) to be applied to the DICOM communication. The buttons at the bottom of the filter list allow filters to be added, removed, moved up in execution order, and moved down in execution order.
- **Conditions** – a list of conditions to check against to decide whether to apply actions. Conditions are checked in order as listed—to change the order click the up or down buttons for the condition.
- **Actions** – a list of actions to perform on the DICOM data if the conditions were true. Actions are applied in order as listed—to change the order click the up or down buttons for the action.
- **Match All / Match Any** – selects whether all conditions in the current filter must match before applying actions or whether only one condition is required to match.
- **OK** – exits the Filter Editor Dialog and saves any changes.
- **Cancel** – exits the Filter Editor Dialog without saving.

**Filter Conditions**

- **DICOM Tag** – condition which checks a DICOM element for certain requirements. **DICOM Tag** allows specification of a DICOM tag and an operator to check against the tag’s value. In this example, if [Patient’s Name] [Contains] [XXX] then the condition is true.

```
DICOM Tag  Patient’s Name  Contains  XXX
```

**Filter Actions**

- **Change Case** – changes the alphabetical case of a DICOM element’s value to either all uppercase or all lowercase. In this example [Patient’s Name] is changed to all [Lower]case.

```
Change Case  Patient’s Name  Upper  Lower
```

- **De-Identify** – provides a sophisticated mechanism to remove and replace certain attributes within a DICOM dataset that may lead to patient identification. For details on use of the De-Identify filter action, see section 5.2, De-Identify Filter Action, below.

- **Composer** – 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. Substitution patterns determine how the output tags are constructed. For details on advanced use of the Composer filter action, see section 5.1, Composer Filter Action, below.

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 which parts of the input elements to combine and how they should be combined. Note that the parts can be used multiple times and may be combined with plain text.

- **Copy** – copies one DICOM element’s value to another DICOM element. In this example the value in [Patient’s Name] is copied over the value for [Patient’s Birth Name].

- **Execute** – executes the C# code contained in the code file provided. This option is provided for special cases where no other actions can perform the needed filtering. If this action is required, please contact Laurel Bridge support.

- **Find/Replace** – checks the value of the DICOM element specified for the given text (or regular expression) and replaces it with the substitute text (or regular expression substitution string). In this example the value in [Patient’s Name] is checked for instances of the text [miner] and any such instances are replaced with the text [minner].
• **Insert/Overwrite** – inserts a new DICOM element or overwrites it if it already exists in the DICOM data set. In this example the [Patient’s Weight] element is inserted into the DICOM data set and given a value of [200]. Any previous value if it existed is overwritten.

| Insert/Overwrite | Patient’s Weight | 200 |

• **Mapping List** – performs a mapping based upon the arguments given. This is very useful if many DICOM data sets need to be changed depending upon a large list of values for particular DICOM elements in those data sets. Arguments for this action are as follows:

  - **Mapping File** – a list of values separated by **Delimiter**. Each row of values should contain as many columns as is specified by the largest Column listed.
  - **Delimiter** – the character used to separate values in the **Mapping File**.
  - **No Match Action** – the action to perform if unable to find a match for the current DICOM data set using the **Match Tag List** selected.
  - **Match Tag List** – a list of tags to match against the current DICOM data set being filtered. These represent the first columns of values in the **Mapping File**.
  - **Replace Tag List** – a list of tags to replace with the values given in the **Mapping File** should a match occur.

In this example, the mapping file (below, `C:\mapping_list.csv`) should contain five columns of values each separated by a **Comma**. If there is no match, the action will **Continue**. The first column of values corresponds to **Accession Number** and the second column of values corresponds to **Study Instance UID**. The third, fourth, and fifth column of values corresponds to **Patient’s Name**, **Patient ID**, and **Study Date**, respectively.

When a new DICOM data set arrives to be filtered the values for the two DICOM elements in the data set described by the first two columns will be compared against every row in the **Mapping File** for a match. If a match is found, then the replacement values in columns 3-5 will be inserted or overwritten in the **Replace Tags List**. For instance, in this example, these DICOM elements will only be overwritten if a match is found.

Here is an example row for this case:

```
1.2.3.4,3.3.1.2.4,JOHN,7.6.5.4.3.2.1,20091215
```

For this row to apply, the DICOM data set passed in must have its **Accession Number** set to “1.2.3.4” and its **Study Instance UID** set to “3.3.1.2.4”. If this matches, then the **Patient’s Name** will be changed to “JOHN”, the **Patient ID** will be changed to “7.6.5.4.3.2.1”, and the **Study Date** will be changed to “20091215”.


• **Move** – moves one DICOM element’s value to another DICOM element. In this example the value in *Patient’s Weight* is moved to the value for *Patient’s Size* even if there is currently a value for *Patient’s Size*. *Patient’s Weight* will no longer exist in the DICOM data set.

![Move](image)

• **New UID** – replaces the UID for the DICOM element listed with a new generated one. In this example the value of [Device UID] will become a newly generated UID. If the DICOM element did not exist previously it will be created.

![New UID](image)

• **Pad** – pads a DICOM element’s value to the desired length with the given character. In this example *Patient’s Name* will be changed to be 25 characters long by inserting the * (asterisk) character on the Left side of the current value for *Patient’s Name* as many times as necessary to pad to 25 characters long. If the current length of the DICOM element selected is already long enough then nothing is changed (the value is not cropped).

![Pad](image)

• **Remove** – removes the given DICOM element from the DICOM data set. If the **Only If Empty** check box is selected, then the element will only be removed if its value is empty or zero characters in length. In this example *Patient ID* is removed from the DICOM data set but **Only If Empty**.
• **Trim** – removes the specified character sequence from the beginning and/or end of a DICOM element’s value, if present. In this example the character sequence “abc” is removed from the beginning of the **Patient’s Name** value. Note that multiple instances of the character sequence could be removed. For example, if the **Patient’s Name** DICOM element contains the value “abcabcSteve”, the resulting value will be “Steve”.

![Trim screenshot](image)

4.1.4. **Client Routing Information**

All client GUI applications have the same interface for specifying routing information.

![Client Routing Information screenshot](image)

• **Client AE Title** – the Application Entity title for the running application (client).
• **Client IP Address** – the IP address to use on the local machine when sending data.
• **Server AE Title** – the Application Entity title of the server machine to which to connect.
• **Server Host** – the host name or IP address of the machine to which to connect.
• **Server Port** – the port number on the server machine to which to connect.
• **Verify** – click this button to send both an ICMP Ping and a DICOM C-Echo test to the server. If logging is enabled, then a log of the C-Echo will be recorded. A green checkmark (✔) means that the verification was successful. A red X (❌) means that it was not successful. A status tooltip can be viewed by hovering the mouse over the checkmark or X.
• **Use TLS 1.0, 1.1, 1.2, 1.3** – enables use one or more SSL/TLS protocols for connection.
• **Ignore Certificate Name Mismatch** – when checked, enables ignoring of name mismatches on SSL/TLS certificates.
• **Allow Self-Signed Certificates** – when checked, enables use of a self-signed SSL/TLS certificate.
• **Send Client Certificate** – when checked, sends the client’s SSL/TLS certificate to the server.
• **Certificate** – the SSL/TLS certificate to send.
• **Password** – the corresponding password for the selected SSL/TLS certificate.
4.1.5. **Server Routing Information**

All server GUI applications have the same interface for specifying routing information and for restricting which clients can connect.

- **Client AE Title** – reject any connecting clients that do not have the specified Application Entity title (select **Allow Any** to accept a client using any AE title).
- **Client IP Address** – reject any connecting clients that are not connecting from the specified IP address (select **Allow Any** to accept a client connecting from any IP address).
- **Client Port** – reject any connecting clients that are not connecting using the specified port number (select **Allow Any** to accept a client connecting using any port number).
- **Server AE Title** – reject any connecting clients that are not attempting to connect to the server with the specified server Application Entity title (select **Allow Any** to accept a client regardless of the AE title it is using for the server).
- **Server Host** – the IP address on the local machine on which to establish the server.
- **Server Port** – the port number on the local machine on which to establish the server.
- **Use TLS 1.0, 1.1, 1.2, 1.3** – enables use one or more SSL/TLS protocol for connection.
- **Ignore Certificate Name Mismatch** - when checked, allows name mismatches in the TLS/SSL certificate to be ignored.
- **Allow Self-Signed Certificates** - when checked, allows TLS/SSL certificates to be self-signed.
- **Allow Missing Certificates** - when checked, allows TLS/SSL certificates to be missing from incoming connections from the client side.
- **Certificate** – the SSL/TLS certificate to use.
- **Password** – the corresponding password for the selected SSL/TLS certificate.

4.1.6. **Transfer Syntaxes for Presentation Context Negotiation**

Some client and server applications provide the ability to select what transfer syntaxes should be included in presentation context negotiation. This ability allows a client application to force DICOM communication to a server to use a specific transfer syntax or to give preference to certain transfer syntaxes. It also allows a server to accept only certain transfer syntaxes from a client or to give preference to certain transfer syntaxes.
All supported transfer syntaxes are initially included in the drop-down box. If the add button is clicked, then the transfer syntax selected in the drop-down box will be moved to the list below. Transfer syntaxes in the list can be removed and moved up or down in priority. The transfer syntaxes at the top of the list will be given greater priority when negotiating presentation contexts and the transfer syntaxes at the bottom of the list will be given lower priority.

4.1.7. Application Window Renaming

All client and server GUI applications can rename the application window title bar, which can be helpful when running multiple instances of the same application.

The renaming dialog is activated by hitting the F2 key (the default Windows file and dialog renaming key shortcut) when the application is open and has focus.
Note that the new name in the title bar only persists until that application instance exits. The next time the application is run it will have the original, standard title in the title bar.

This renaming feature is also available when starting any of the GUI applications that support renaming at the command line and is particularly helpful when scripting the startup of several collaborating applications. This is done using the -name command line switch. An example of starting up one renamed Store Server and three renamed Study Generator instances is provided in the demo batch script example_gui_renaming.bat in the PowerTools installation directory.

4.1.8. External Data Dictionary Management

The Dicom File Editor and Dictionary Finder applications provide access to functionality that selects and manages an external data dictionary. When activated via an options setting, a status indicator appears in the upper right of the main application window.

When the indicator is visible, clicking on it opens the external data dictionary manager, where the user is presented with four choices of the Type of dictionary to load:

- **None** – do not load any external data dictionary.
- **Blank** – create an empty text file and load it as the external data dictionary.
- **Built-in DICONDE Dictionary** – load a standard DICONDE data dictionary.
- **User-selected Dictionary** – enable the selection of a data dictionary file.

When the User-selected Dictionary option is selected, the file browser is activated so the user can select a data dictionary File.
Buttons on the Manage External Data Dictionary dialog do the following when clicked:

- **Reset** – sets the selected Type to None and clears the File entry field.
- **Show Folder** – opens the default application folder containing the current and any backup external data dictionaries.
- **OK** – saves the current options for external data dictionary. If the data dictionary in use was changed, a backup copy of the previous data dictionary is created in the same directory.
- **Cancel** – cancels any changes made to the selected options.

4.2. **DICOM Client Applications**

PowerTools provides several GUI-based DICOM client applications which connect to a DICOM server on a remote machine and transmit DICOM communication. The advanced functionality of these applications is discussed below. Other console-only DICOM client applications are described in Section 6.

4.2.1. **Query Client**

Query Client can query a query SCP containing a database of DICOM images and request information about certain images. It has both console and GUI versions.
The main screen of the Query Client application.

Using the command-line interface, query parameters can be specified either directly using the \(-q\) flag, or via a configuration file. For example, a command line query for all patients that have a Patient ID of “112233” and that returns those values plus Accession Number and Referring Physician looks like this:

\[
\text{QueryClient [other params] } -q \ 0008,0050= -q \ 0008,0090= -q \ 0010,0020=112233
\]

Instructions for using a configuration file to perform queries are contained in the example configuration file (\(\text{QueryClientExample.cfg}\)) found in the PowerTools installation directory.

Query Client enables configuration of the query Type & Method. These settings include three query modes, Find, Move, and MWL described by Query Type, and two search methods, Relational and Hierarchical, described by Query Method.

Find mode queries a query SCP for DICOM data sets which contain DICOM elements which match the tags selected in the GUI. Tags with an empty value will be populated upon response from the server. Relational and Hierarchical search methods support queries to PACS of the corresponding query architecture.
**Move** mode queries a query SCP in the same way, but instead of returning the found DICOM elements the entire DICOM data sets which match are sent to the Move destination, described by the **Move AE Title** (the query server should have a mapping for this AE title so that it knows where to send the DICOM images). **Relational** and **Hierarchical** search methods support queries to PACS of the corresponding query architecture, and **Range Match** determines whether or not extended range matching negotiation is requested.

![Query Interface](image)

**MWL** mode enables a mechanism for performing **Modality Worklist** queries using the **Query Client** user interface. When the application is in MWL mode, **Modality Worklist** queries of arbitrary depth can be crafted in the **Query Tag List**.

![Query Interface](image)

Other key components of the **Query Client** user interface not fully described above include:

- **Range Match** – determines whether extended range matching negotiation ([Combined Date/Time](#)) or standard **Separate Date/Time matching** will be used for Date and Time attributes with a **Find** query. The **Range Match** value does not apply to **Move** or **MWL** queries.
- **Root** – the root, or information model, of the query server (Study, Patient, or PatientStudy).
- **Query Level** – the level at which to query on the server (Patient, Study, Series, or Image).
- **Query Tag List** – a list of tags with which to query the query SCP. Values can be edited to request a match, and if left blank that value will be filled in upon return.
The green plus and red minus buttons are used to Add a tag to the list or to Remove the selected item, if any, from the list of items. The green up and down arrows move the selected item up or down in the list.

When in MWL mode, when the Query Tag List can be in a tree layout, the Add button will add a tag as a child of the currently selected item in the list, or at the top level of the list if no item in the list is selected. In Find or Move mode, the Add button simply adds a tag to the list.

In all modes, the Remove button will immediately remove any item that has no child items. In MWL mode, the Remove button will cause a prompt for confirmation before removing a selected item and all of its child items if that item has one or more child items.

A context menu pops up when an item in the list is right-clicked. Depending on the current mode, an item can be added, added as a child of the selected item, removed along with any child items, or the entire list can be cleared.

- Use Default Tags – populates the list with default query tags appropriate to the currently selected mode. Find/Move queries have a Minimal and Standard set of defaults, while MWL queries have Minimal, Standard and Extended default sets.
Predefined default tag sets are available. (two for Find/Move, three for MWL).

- **Find/Move/MWL Query** – clicking this button initiates a C-Find, C-Move or Modality Worklist query operation to the query SCP using the specified information.

![Options dialog](image)

The **Options** dialog in the **Query Client** provides Logging and Filtering support, as well as a convenient interface to configure **Query Timeout** and to **Limit Max Query Results** returned. **Clear application log before each query** starts with a fresh log. The **Show Retrieve AE Title** and **Show Instance Availability** options enable the display of the respective information in the query results.

- **Query Timeout** – the amount of time in seconds to wait before timing out the query (times out after no response has been received for the number of seconds specified).
- **Limit Max Query Results** – if checked then the maximum number of queries returned is limited to the value specified.

In addition to the **Options** dialog, the **Tools** menu enables the ability to **Add Calculated Attribute Tags to Query**. When selected, the list of query Tags will include the following:

- Number of Patient Related Studies
- Number of Patient Related Series
- Number of Patient Related Instances
- Number of Study Related Series
The PowerTools **Query Server** supports the listed calculated attribute tags, which are calculated on-demand for each query received by traversing a directory structure. Note that the Alternate Representation Sequence tag is not yet supported. Typical usage pattern for these tags in the **Query Client** application is to add them all and then selectively remove any that are not desired in the current query.

The query results table provides a variety of sorting and selecting capabilities.

- Sort values in a column - click on the column heading
- Reverse sorting direction in a column - click on the same column heading
- Select all values in a row - click on the **Result #** row heading for the row
- Select all values in a column - hold the Shift key and click on the column heading
- Select multiple columns or rows - hold the Shift key and click on multiple column headings or **Result #** row headings
- Select a rectangle of result cells - click on a cell and drag to highlight a rectangular region of cells
- Select arbitrary result cells - hold the Ctrl key and click on multiple cells to select them
- Select all results - hold the Shift key and click on the **Result #** column heading
- Reset sorted order to original and deselect everything - click on the **Result #** column heading
- Copy selected values - right-click to bring up the context menu and select **Copy Selection to Clipboard**, or type Ctrl-C to copy selected values
4.2.2. Store Client

The **Store Client** is used to perform DICOM image store operations and has both console and GUI versions.

Once DICOM images have been selected and the **Send** button is clicked, images are sent to the selected DICOM store server. Sending can be canceled at any time by clicking the **Cancel** button. Upon completion, an optional status report of the completed store operation can be viewed. Advanced settings in the **Options** dialog of **Store Client** are described below.

- **Continue Storing After Image Read Or Store Error** – if checked and an error occurs during the transmission of a single DICOM image, then any unsent images will continue to be attempted to be sent.
- **Fail On Rejected Presentation Context** – if checked and presentation context negotiation is unsuccessful for an image then the store attempt will fail.
- **Use Separate Association For Each Image** – if checked then each image will be sent using a separate association. By default, all images are sent together in a single association. Enabling this option can be useful for load testing of a server or network.
- **Delay each store by** - optionally define a delay to be used between each image being sent, from 0 to 99.999 seconds.
• **Hide Report On Successful Send** – If all images are sent successful and this is checked then the status report shown at the conclusion of the send will not be displayed.

• **Context Mode** – the mode with which to negotiate presentation contexts using the Desired Client Transfer Syntaxes For Presentation Context Negotiation List (section 4.1.6).
  - **As Encoded Transfer Syntax Only** – sends all images in their current transfer syntax. If the SCP will not accept an image’s transfer syntax, then that image will not be sent.
  - **As Encoded Else SCP Preferred Transfer Syntax** – attempts As Encoded Transfer Syntax Only, and if that fails then the image will be sent in whatever transfer syntax the SCP prefers.
  - **As Encoded Else SCU Preferred Transfer Syntax** – attempts As Encoded Transfer Syntax Only, and if that fails then the image will be sent using the desired client transfer syntaxes for presentation context negotiation list if possible (the transfer syntaxes at the top of the list will be given a higher priority).
  - **SCP Preferred Transfer Syntax Only** – images will be sent in whatever transfer syntax the SCP prefers.
  - **SCU Preferred Transfer Syntax Only** – images will be sent using the desired client transfer syntaxes for presentation context negotiation list if possible (the transfer syntaxes at the top of the list will be given a higher priority).

4.2.3. **Study Generator**

The Study Generator uses customizable Patient Name, Date of Birth, Sex, Modality and Study Date attributes to generate and store via DICOM DIMSE or DICOMweb STOW transfer mode one or more single-frame or multi-frame DICOM images to a specified Store Server profile, comprised of a transfer mode specific combination of Host, Port, Called AE, Calling AE, Transfer Syntax, and selected STOW Options. The application supports defining up to 32 of these Store Server profiles to simplify repeated testing on multiple destinations.

Hovering the mouse over the Modality produces a tooltip that displays the modality name, element name and SOP Class UID for the element.

Any number of additional DICOM tags (Extra Tags) can be specified, and the user can specify the Number of Studies, Series per Study, Images per Series, Frames per Image, and the number of Rows and Columns in each frame, and whether to Use Separate Association per Study, Series or Image. Clicking on the Generate button starts the store job, which can be canceled at any time by clicking the Cancel button.

If the Store Studies in Parallel option is checked, the Study Generator will attempt to concurrently store each of the requested Number of Studies, up to a limit determined by the Windows operating system. There also may be an even lower limit for the maximum number of
concurrent stores (i.e., associations) depending on the specific store server being targeted. The application attempts to warn the user when the operating system limit is being exceeded.

An optional Background Image can be selected that will be used as the image content for all generated images. When present, the background image will contain a visible watermark indicating that the image is a “SAMPLE.” When not present, a solid black background will be used. An optional textual Tag Overlay of study information also can be displayed.

Any image in one of the more common image formats (JPEG, PNG, BMP, GIF and TIFF), as well as a DICOM image file that contains a viewable image, can be used as a Background Image for generation. A selection of example images for a variety of modalities is in the backgrounds subfolder in the PowerTools installation directory. Note that when a Modality is selected, a relevant image for that modality is automatically set as the Background Image.

If the Extra Tags for Columns and Rows are loaded after the Background Image is selected, their values will be auto-sized to the dimensions of the Background Image. See below for information on the Load Extra Tags options menu.

In addition to Extra Tags, a DICOM file Template can be selected, like the console version of the Study Generator application (described below). If selected, a Template will be used to populate the generated DICOM data set. The interaction of DICOM element tags defined in various places in Study Generator works like this:

- The Template, if selected, is used first to populate the generated DICOM data set.
- Next, any specified Study Information tag values are added to the data set, potentially replacing identical tags defined in the Template.
- Finally, the Extra Tags are added to the data set, similarly replacing identical tags already defined in the data set.

When a file is selected as a Template, right-clicking the filename displays a context menu with the option to Clear the Template or Copy its complete file path to the system clipboard.
One or more Store Server Profiles can be defined, with the list of profiles each specified by a Transfer Mode of DIMSE, STOW Part 10, STOW JSON, or STOW XML that determines which of the other settings in a profile are used. Note that once defined, these profiles persist across uses of the Study Generator. Common to all transfer modes are Host, Port, and Transfer Syntax. DIMSE-specific settings are Called AE, Calling AE. For STOW transfer modes, STOW Options are enabled to allow configuration of Resource Path, Batch Size, and Response Type. Each of the profile settings has pop-up tooltips that provide more information about the setting.

The STOW JSON and STOW XML transfer modes are limited to single-frame image data. If the number of Frames/Image is greater than 1 when using one of these two modes, an information icon and message will be shown noting that only the first frame will be sent. The Image Size and Total Data fields are updated to reflect this difference, as well.

To add a new store server profile, click the button and enter a new profile name into the dialog:
Once a profile has been added to the list of profiles, its values can be directly edited in the main GUI screen by first selecting the desired profile name from the list and then editing the four values underneath. Note that when adding a profile, all existing settings for the currently selected profile (Host, Port, Called AE, Calling AE, Transfer Syntax, Transfer Mode, and STOW Options, regardless of which are enabled) will be copied into the newly added profile. Unlike many other PowerTools clients, note that the use of SSL/TLS is not yet supported in Study Generator.

To delete a store server profile, select the desired profile in the list and click the button, and confirm deletion of the profile.

There is no direct way to rename a store server profile. To effectively rename a profile, select the profile to be renamed (e.g., “Old Profile”), add a new profile (e.g., “New Profile”) with the desired new name, and then delete the first profile (i.e., delete “Old Profile”).

The Tools menu provides access to the Options dialog and to a variety of time-saving shortcuts.
• **Tools**
  o **Options** – launches the Options Dialog (see below).
  o **Load Extra Tags** – provides five convenient selections that will populate the Extra Tags list with the indicated tags. Note that a tooltip will pop up for each of these selections, providing information about the tags that will be added.
  o **Clear Extra Tags** – clears all tags from the Extra Tags list.
  o **Revert To Factory Settings** – resets the application to its initial settings.

When loading the **Bits Allocated and Stored** tags, the user can indicate generation of 8 or 16 bit images by setting the appropriate value in the **Bits Allocated** tag. If not indicated, 8 bit images are generated.

The **Options dialog** in the **Study Generator** provides support for **Logging**, as well as several additional settings.

![Options Dialog](image)

By selecting **Show group & element numbers in Extra Tags list**, the group and element numbers are displayed in the drop-down list of Extra Tags to support locating tags by group and element number in addition to tag name.
Selecting **Delay each** enables the store of each Study, Series or Image to be delayed by a configurable number of seconds between each store operation. Each store can be delayed by exactly or up to a desired number of seconds, from 0.0 to 99.999 seconds.

Selecting **Retry on error after** enables a limited number of retries (maximum is 99 retries) with a delay of a configurable number of seconds (maximum is 999 seconds) between each attempted retry.

Selecting **metric** or **binary** units for **View frame size, image size, and total data in** causes the display of those three values to use metric notation (e.g., MB or Megabytes = 1,000 x 1,000 bytes) or binary notation (e.g., MiB or Mebibytes = 1,024 x 1,024 bytes).

**DIMSE Timeouts** for **Send** and **Receive** can be configured in the range 0 to 999 seconds. The **STOW Timeout** can be configured in the range 0 to 999 seconds, with a timeout of 0 indicating an infinite timeout.
Selecting Run generator in fast mode enables a “fast” mode of generation to improve performance for situations where the fastest generation possible is needed. In the fast mode, a single generic image is generated and used over and over, and only the specific DICOM tags that change with each image are updated. Load tests using fast mode have shown an approximate speedup of 300% (3 times faster) as compared with standard mode generation.

When started at the command line, the GUI version of the Study Generator has a -run switch that causes the generator to immediately begin running upon application start up.

The console version of the Study Generator makes use of a configuration file that controls generation of a variety of Patient, Study, Series and Image tags. Several built-in functions support generation of random values for UID, date, time, name and gender. Detailed instructions are contained in the example configuration file (StudyGeneratorExample.cfg) found in the PowerTools installation directory, along with a small, example DICOM file (mr-knee.dcm) that can be used as a template for generating studies.

4.2.4. Verification Client

The Verification Client connects to a remote server and checks to see if it is currently running (both ICMP ping and DICOM C-Echo tests are performed).

Its verification functionality is available in every client application, visible as a Verify button on the server configuration portion of the client GUI window. Optional settings for SSL/TLS enable the use of a secure connection. Upon completion, status about the numbers of request sent and the elapsed time is displayed. In addition to standard logging and filter support, the Options dialog enables selection of a Context Mode and a list of Desired Client Transfer Syntaxes For Presentation Context Negotiation like the support in the Store Client (section 4.2.2).
4.3. DICOM Server Applications

PowerTools provides several DICOM server applications which receive connections from a DICOM client on a remote machine. The more advanced functionality of these applications is discussed below. Other console-only DICOM server applications are described in Section 6.

4.3.1. Store Server

The **Store Server** provides DICOM SCP functionality with optional DICOM Storage Commitment service functionality. Received images are displayed in a grid display, with each row in the grid representing all images received within the same association. Rows can be expanded to show a sequence of cards displayed, each with a summary of DICOM header tags and a thumbnail view of the image, if any.

The **Store Server** is started and stopped (or paused) using the run button ![run button](image) and the stop (or pause) button ![pause button](image). Received images either can be immediately discarded or saved to a folder by checking **Write Images To Disk** and specifying an existing path. In addition, typical server routing and TLS/SSL options can be configured.

Metrics displayed for each association are: **Start Time**, **End Time**, **Duration**, **Called AE Title**, **Calling AE Title**, **Calling Presentation Address**, **Image Count**, and **Status**. A calculated **seconds per image** tooltip is displayed when the mouse is hovered over any **End Time** or **Image Count** value.

Within each card, if an image contains multiple frames, the **Number of Frames** is displayed. When the thumbnail image is clicked, it will be opened in the **Dicom File Editor** for viewing and editing.
Main user interface screen of the Store Server.

Clicking on a thumbnail image will open the underlying DICOM image file in the Dicom File Editor.
The Number of Frames in a multi-frame image is shown.

For DICOM image files that do not contain viewable image data, a “No Image” indicator is displayed on the Image Thumbnail:

The No Image indicator.

Large numbers of image and log files can be generated in the process of running the Store Server. To assist with file system cleanliness, when the application is exited the user is presented with a dialog offering the option of deleting these image and log files before exiting.

If the Don't ask me again checkbox is checked, this dialog will not appear again unless the feature is reset in the Options dialog.

The Clear All Before Exit dialog of the Store Server.
Advanced settings in the **Options** dialog of the **Store Server** related to routine storage functionality are described below.

- **Automatically start server on application start** – if checked then the server will attempt to start immediately upon application startup.
- **Don’t ask to Clear All on exit** – if checked then the “Clear All Before Exit” dialog will not appear on application exit.
- **Keep most recent association in list visible** – if checked then the association list automatically scrolls so the most recently received association is always visible.
- **Limit list** - maintains a maximum number of associations in the list to prevent the list from growing exceedingly large.
- **Favor SCU proposed presentation contexts** – if checked then the connecting SCU will be given preference when negotiation presentation contexts. If unchecked, then Store Server will negotiate presentation contexts based upon its **Supported Server Transfer Syntaxes For Presentation Context Negotiation List**.
- **Manage Supported SOP Classes** – when clicked, the **Manage Supported SOP Classes** dialog opens, enabling the user to add, remove and modify the current list of supported SOP classes **Store Server** uses.

The **Store Server** also supports Storage Commitment request handling when enabled. The **Options** dialog has the following Storage Commitment support:

- **Enable Storage Commitment** – if checked then the server will handle incoming Storage Commitment requests using the following corresponding settings.
- **Store Server AE Title** – specifies the calling AE title that the Store Server uses when it sends a Storage Commitment N-EVENT-REPORT notification as an SCU.
- **Delay response by** – if checked it specifies the number of seconds to delay sending of a response to a Storage Commitment request. Delay can be from 1 to 999 seconds. Unchecked is the same as a 0 second delay. A delay of -1 indicates that a response will not be sent to support manual, asynchronous responses (see below).
• **Use same association as request** – future feature is under development to support sending a Storage Commitment response via the same association on which a request is received.

• **Remote AE Title** – specifies the called AE title for the remote Storage Commitment SCU acting as a server for N-EVENT-REPORT notifications.

• **Remote Host/IP** – specifies the host name or IP address of the remote Storage Commitment SCU acting as a server for N-EVENT-REPORT notifications.

• **Remote Port** – specifies the port number of the remote Storage Commitment SCU for N-EVENT-REPORT notifications.

The **Manage Supported SOP Classes** dialog enables the user to selectively delete any, or all, of the UIDs in the list of supported SOP class UIDs and names. Additional SOP classes can be added by entering a new **UID** and **Name** and then pressing the green add button. Pressing the **Reset** button restores the default list of supported SOP classes.

![Manage Supported SOP Classes dialog](image)

The **Manage Supported SOP Classes** dialog.

In addition to the **Clear All Before Exit** functionality that can be configured via the **Options** dialog, the list of associations can be cleared via the **Clear All** item in right-click context menu, the **Clear All** item in the **File** menu, or the menu item keyboard shortcut **Ctrl-Delete**. One or more selected items in the association list can be deleted by selecting the items you wish to remove and then either hitting the **Delete** key or using the context menu **Remove** item. Note that **Clear All** and **Remove** are not available while the Store Server is **Running** and **Write Images To Disk** is being used to save incoming images.

![Context menu of the Store Server](image)

The right-click context menu of the **Store Server**.
When started at the command line, the GUI version of the Store Server has a `-run` switch that causes the generator to immediately begin running upon application start up, which is the same optional functionality as that described above.

The Transfer Syntaxes for Presentation Context Negotiation setting (section 4.1.6) that is common in a number of client and server applications has been extended in the Store Server with the ability to add one or more custom transfer syntax entries to enable graceful handling of a wide variety of non-standard and incorrect transfer syntax values.

- **Add Custom Transfer Syntax UID and Name** – by selecting this option in the drop-down list and clicking the add button, the user can enter any desired transfer syntax UID and Name. The list automatically prevents duplicate or blank UIDs from being entered.

![Extended Transfer Syntax settings management in Store Server.](image)

When Storage Commitment support is enabled via Options dialog, additional information is displayed in the grid of received data. The following statuses can be displayed for Storage Commitment entries:

- **SC: Acknowledged** - a Storage Commitment request has been received but has not yet responded to.
- **SC: Report Sent** - a Storage Commitment response (an N-EVENT-REPORT notification) was successfully sent.
- **SC: Send Report Failed** - a Storage Commitment response (an N-EVENT-REPORT notification) failed (reason is logged).

4.3.2. Verification Server

Verification Server can be used to provide a very simple Verification SCP to test a verification SCU. It is configured using the common Server Routing Information element (section 4.1.5). As with other PowerTools server applications, its Options dialog is only available when the server is not running.
Typical server routing and TLS/SSL options can be configured. The functionality of the Verification Server is present in every server application and will not be described further.

4.4. DICOM Data and File Manipulation Applications

PowerTools provides DICOM applications which modify one or more DICOM files on disk as well as applications used to manage DICOM data. The advanced functionality of these applications is discussed below.

4.4.1. DICOMDIR Creator

DICOMDIR Creator takes as arguments an Image Directory where DICOM images are located (may also be contained in subdirectories) and a DICOMDIR File Name (usually DICOMDIR). The application then creates a directory/image hierarchy from the images found in the image directory and creates a DICOMDIR file using the information gathered which is placed in the Image Directory upon completion. Results are shown in the Results list.
Also, please note the following when creating a DICOMDIR:

![DICOMDIR Creator](image)

4.4.2. DICOM File Editor

DICOM File Editor provides a three-tabbed view for inspecting DICOM files: a tabular view of the header contents, a hex viewer, and an image viewer for the pixel data. It is recommended to set the DICOM File Editor as the default viewer for .DCM files in Windows to enable faster viewing/editing of image files. Some advanced settings for this application are described below.

The DICOM File Editor menu bar has several options unique to the application.
File Menu.

- **File**
  - **New** – creates a new DICOM data set for editing.
  - **Open...** – opens a DICOM image or DICOMDIR file from disk. This functionality can also be performed by dragging a DICOM image or DICOMDIR file from Windows Explorer into the main view of the DICOM File Editor.
  - **Open Folder...** – opens a folder of DICOM files from disk. This functionality can also be performed by dragging a folder containing DICOM image or DICOMDIR files from Windows Explorer into the main view of the DICOM File Editor.
  - **Recent Files** – provides a convenient list of recently opened files.
  - **Import From > LBS CFGGroup...** – similar to **Open** but uses a Laurel Bridge data file instead of a DICOM file.
  - **Export To > LBS CFGGroup...** – saves the currently open DICOM data set as a Laurel Bridge data file.
  - **Export To > Text Document...** – dumps the currently open DICOM data set to a text file in text format.
  - **Save** – saves the current DICOM data set to a file.
  - **Save As...** – saves the current DICOM data set to a file specifying the file name and transfer syntax in which to save the image.
  - **Save As PDF** – saves the current DICOM data set, including the first image frame if there is one, to a PDF document in one of two formats. Saving With All Header Tags includes the entire set of DICOM header tags in the PDF. Saving With
Demographic Header Tags includes only a minimal set of DICOM header tags about the patient, study, series, instance.

- **Save Image As** – saves the currently displayed image frame on the Image panel as a regular image (BMP, GIF, JPEG, PBM, PGM, PNG, PPM, TIFF). For multi-frame images, the current frame will be saved to one of these formats. Additionally, for multi-frame images only, the complete multi-frame image can be saved as an Animated GIF at a 10 fps rate.
- **Save Header As Text** – saves the header of the current DICOM data set as a plain text file, specifying the file name of the saved file.
- **Save Hierarchy As PDF** – saves the current study hierarchy displayed in the left panel as a PDF.
- **Save Encapsulated PDF As** – saves an Encapsulated PDF element to a file if one is present in the current DICOM data set.
- **Save MPEG Pixel Data As** – saves the pixel data element, if present and determined to be MPEG data, of the current DICOM data set to an MPEG file.
- **Print Preview**... – shows a preview of what will be printed if the current DICOM data set (without pixel data) is printed.
- **Print**... – prints the current DICOM data set to a printer (without pixel data).
- **Close** – closes the currently open data set, if any.
- **Clear** – clears the list of DICOM data sets in the navigation view.
- **Exit** – exits **DICOM File Editor**.

<table>
<thead>
<tr>
<th>Tools Menu</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Options...</td>
<td>Ctrl+T</td>
</tr>
<tr>
<td>Find in Open DICOM Data Set...</td>
<td>Ctrl+F</td>
</tr>
<tr>
<td>Clear Find Highlighting</td>
<td>Ctrl+X</td>
</tr>
<tr>
<td>Filter Open DICOM Data Set...</td>
<td>Ctrl+Shift+F</td>
</tr>
<tr>
<td>Copy Path of Open File to Clipboard</td>
<td>Ctrl+C</td>
</tr>
<tr>
<td>Revert To Factory Settings</td>
<td></td>
</tr>
</tbody>
</table>

- **Tools**
  - **Options** – launches the Options Dialog (see below).
  - **Find in Open DICOM Data Set...** – launches the Find dialog that enables search operations in the open data set in both the header view and hex view panels.
  - **Clear Find Highlighting** – clears the highlighting that results from a Find All operation in the Find dialog after a search in the header view panel.
  - **Filter Open DICOM Data Set** – launches the Filter Editor Dialog and applies any filters to the currently open DICOM data set.
  - **Copy Path of Open File to Clipboard** – copies the complete path of the open DataSet file to the system Clipboard.
  - **Revert To Factory Settings** – in addition to the default reset behavior, selecting this will clear the Recent Files list and the list of recent search terms in the Find dialog.
The **Options Dialog** in **DICOM File Editor** allows the specification of several unique settings.

![Options Dialog](Image)

- **Enable Logging** – Checking this option will enable logging for **DICOM File Editor** use. For more description of logging, see the description of the Options Dialog in section 4.1.2.
- **Value Separator Character(s)** – specifies the character(s) to use to separate multi-valued elements in the main value display.
  - **Comma and Backslash** – a multi-valued element will appear as [a/b] or [1,2]
  - **Newline** – a multi-valued element will place each new value on a new line.
- **Save Files in Ch 10 Format** – when checked, data sets will be saved in Chapter 10 format.
- **Invent Group 2 Data** – when checked, group 2 data will be invented if needed.
- **Read Invalid VR as Implicit** – an advanced and rarely needed option, it enables data sets containing a variety of errors to be opened and viewed.
- **Items shown in Recent Files List** - determines how many recently opened files will be listed in the Recent Files menu item.
- **Show external data dictionary indicator** – when checked, the external data dictionary indicator is displayed on the menu bar (section 4.1.8).
- **Manage** - when clicked, the **Select External Data Dictionary** (section 4.1.8) dialog pops up, enabling selection and management of external data dictionaries.
• **Show names for well-known UIDs** – causes the names of any well-known UIDs to be displayed in the header view tab rather than the UID strings themselves. The original UID will be displayed in a tooltip when the name is hovered over with the mouse.

• **Display UN element values as text when possible** – If checked, examines bytes in UN elements and displays them as text if all are ASCII printable characters.

• **Expand DICOMDIR items** - when checked, DICOMDIR items in the Navigation List will be expanded by default.

• **Expand Patient items** - when checked, Patient items in the Navigation List will be expanded by default.

• **Expand Study items** - when checked, Study items in the Navigation List will be expanded by default.

• **Expand Series items** - when checked, Series items in the Navigation List will be expanded by default.

The **Find in Header** Dialog for the header view panel of the **DICOM File Editor** provides searching within the DICOM header values of the open data set. The “Find what” field maintains a list of recently used search term. There also is a **Find in Hex** Dialog for the hex view panel (see below). When switching between the header, hex and image view panels, any open **Find** Dialog remembers whether it is open and if so, it automatically closes and reopens as its panel is left and returned to.

[Image: Find in Header dialog]

Clicking **Find Next** will search in the selected search **Direction** (Up or Down), while clicking **Find All** will highlight all matching elements in the data set. Selecting **Match whole word** will limit search results to matches that have whitespace or are at the start or end of a field rather than an arbitrary partial match. Selecting **Match case** will make the search case-sensitive.
Highlighting after a Find All operation for the search term “study”.

When Use wildcards is selected, a limited set of wildcard notation is available to enable more powerful and flexible searching. Selecting the help icon next to Use wildcards opens the Using Wildcards dialog that contains a summary of the wildcard syntax.

The navigation view on the left side of the UI displays the four levels of the imported instances: patient/study/series/instance. The displayed values for each level are: Patient ID, Study Instance UID, Series Instance UID, and SOP Instance UID. All studies having the same Patient ID will be displayed underneath that entry. All series having the same Study Instance UID will be displayed underneath that entry. All instances having the same Series Instance UID will be displayed underneath that entry.
The header view of the **DICOM File Editor** lists the DICOM elements loaded from the selected DICOM file in the navigation area. Each element can be edited by double clicking the element (except for sequence items). Also, right clicking any element in the display will provide a different context menu depending upon the type of element selected.

Hovering over the **Length** of any element in the header view panel will cause a descriptive tooltip to pop up, reporting on the total number of bytes stored in the file for that element, including the number of bytes in its header.

**Institution Name** is highlighted in the header view panel.

A tooltip reports bytes stored for each element.
When hovering over the VR of an element, a tooltip describes the meaning of the VR.

For Pixel Data elements that are stored in a compressed format, their Length is correctly reported as -1, and the Length field is highlighted. Pixel Data element data size information is reported in a tooltip.

It is possible that a file containing compressed Pixel Data claims to store its data as OW, even when the data is OB. In this case, the VR field is highlighted and a tooltip explaining the discrepancy can be shown. This discrepancy can be confirmed using the hex view panel.

When an element is clicked on in the header view panel, the corresponding bytes for that element are highlighted in the hex view panel. Similarly, when some byte is clicked in the hex view panel, the element that contains that byte (and the bytes adjacent to it) is highlighted in the header view panel. When an element is highlighted in one image, and a different image is selected for viewing, the same element in the newly selected file will be highlighted if present.
Hexadecimal byte values corresponding to the Institution Name element are highlighted in the hex view panel.

Right-clicking on highlighted bytes in the hex view panel enable those bytes to be copied to the Clipboard, both as binary values and text values, for pasting into applications that accept either binary or text.

Highlighted bytes can be copied to the Clipboard.

When any byte in the DICOM preamble (for which there is no corresponding DICOM element) is selected, the entire preamble is selected in the hex viewer panel and a descriptive tooltip is displayed.
The **Find in Hex** Dialog for the hex view panel of the **DICOM File Editor** provides searching within the hex values of the open data set. The “Find what” field limits the search term to valid hexadecimal digits.

Clicking **Find Next** will search forward from the current position in the hex view panel. If matching hex values are found, they are highlighted in the hex and string value areas and the starting address of the match is displayed on the dialog.

When an Encapsulated PDF document is present in the current data set, it can be saved to a file and viewed from either the File menu option or by right-clicking on the Encapsulated Document element in the header view panel and selecting **Save Encapsulated PDF** or **View Encapsulated PDF**.
When present, DICOM image data is displayed with the built-in basic window-level image viewer on the image panel. The displayed window width and window level can be adjusted using the corresponding sliders or by dragging the mouse. The levels are reset by clicking the green reset arrow button in the lower right. Right-clicking the green reset arrow button enables **Auto Window Level** to be performed by applying a histogram-based analysis to the displayed image.

When displaying a multi-frame image, the current and total number of **Frames** is displayed. Clicking the play button icon causes the sequence of frames to be animated. The mouse wheel can be used to move forward and backward through the sequence one frame at a time.
Hovering the mouse over the header, hex or image panel tabs will cause the currently displayed DICOM file’s complete path to be displayed in a tooltip.

Right-clicking on any of the three tab headers, or anywhere on the Image panel, will cause a context menu to pop up, enabling the currently displayed DataSet file to be shown in its folder on the file system (Show in Folder), saved as a regular image (BMP, GIF, JPEG, PNG, TIFF) (Save Image As) or have the path of the DataSet file copied to the system Clipboard (Copy File Path to Clipboard).

When an element has an invalid value, it is displayed with red highlighting (below). Specifically, this is an indication that the VR validator has failed and that the element value has an invalid length.

The context menu for each type of element is shown and described below.
• **Add Element** – adds a new element to the DICOM data set (standard element or sequence element).
• **Edit Element** – edits the currently selected element (sequence items cannot be edited).
• **Delete** – deletes the currently selected element(s) or item(s).
• **Copy Value(s) to Clipboard** – copies one or more selected items to the system Clipboard.
• **Add Sequence Item** – adds an item to the selected sequence element.
• **Edit Sequence Element** – edits the selected sequence element.
• **Expand All** – expands all sequences for complete viewing.
• ** Collapse All** – collapses all sequences for compact viewing.

The **DICOM Element Editor** provides an interface for editing an individual item in a DICOM data set. Its fields are described below.

• **Tag Code** – the group/element hexadecimal value describing the DICOM element (private tags may be specified by typing them in directly).
• **Tag Name** – the name of the DICOM element.
• **VR** – the value representation type of the DICOM element.
• **VM** – read only field describing the value multiplicity (usually “1” but will be greater for multi-valued elements).
• **Value(s)** – the values for the DICOM element with each value on a new line.
• **Generate UID** - only present for UI elements. Click to generate a new UID in the **Value(s)** field. The generated UID will be inserted at the cursor position, and will replace any portion of the value that is highlighted.
- 54 -

- **Validate** – validates the DICOM element to make sure all fields are DICOM valid.
- **OK** – saves the element and exits the dialog.
- **Cancel** – exits the dialog without saving.

### 4.4.3. DICOM Filter

**DICOM Filter** allows for the filtering of multiple DICOM image files on the disk. For example, if a patient’s name was spelled incorrectly in an entire study of hundreds of images, simply select the directory containing the images and add a filter to correct the name. The functionality in this application is shared with other applications and will not be described further here.

![Main user interface of DICOM Filter application.](image)

### 4.4.4. Dictionary Finder

**Dictionary Finder** provides the ability to search the DICOM data dictionary for a particular UID (unique identifier) or for a specific DICOM element. Settings for this application are described below.
There are four different dictionaries available for searching: **SOP Classes**, **Transfer Syntaxes**, **Attribute Tags**, and **VR Names**. Once one of these dictionaries has been selected by clicking on one of the four tabs, relevant columns are displayed and are searchable by typing into the **Filter** text box.

Search results consist of only the rows in the table of values that match the text typed into the **Find** text box, subject to one of four matching criteria:

- **Contains** – the find text is contained anywhere in a field.
- **Starts With** – the find text is found at the beginning of a field.
- **Ends With** – the find text is found at the end of a field.
- **Wildcards** – the find text can contain wildcards that enable a simplified regular expression style matching.

When Wildcards mode is selected, a help icon is enabled.

When the help icon is clicked, a **Using Wildcards** instruction dialog pops up. A description and screen capture of the **Using Wildcards** dialog is contained above in the **Dicom File Editor** description (section 4.4.2).

The **Dictionary Finder** provides access to loading and managing an external data dictionary (section 4.1.8). The external data dictionary indicator can be shown or hidden using the **Show External Dictionary Indicator** option in the **Tools** menu.
4.4.5. PDF to DICOM

The PDF to DICOM tool enables one or more existing PDF files to be converted to DICOM Encapsulated PDF files, which optionally can be written to files on the local disk or stored to a PACS. A sequence of screens guides the user step-by-step through the conversion process. The user first selects one or more PDF files to encapsulate. Next, a data entry step allows DICOM tag values to be customized.

Finally, the user selects an output directory to which the DICOM Encapsulated PDFs will be written, or optionally the Hostname or IP address, Port number, Called AE and Calling AE of a PACS to store the files.

4.4.6. Structured Report Extractor

The Structured Report Extractor is a tool that parses one or more DICOM Structured Report files. It is designed specifically to generate a mapping file for use with the Laurel Bridge Software Compass™ product and third-party integrated dictation and transcription systems though it can be used more generally to reveal the fields and values contained in one or more DICOM Structured Reports. It can also be used to identify problems in a mapping file.
The **Mapping File** field is the path to a text file containing the **Structured Report Search Term** values and their corresponding mapped **Custom Mapping Field** and **Rounding Precision** values.

The **Structured Report(s)** field contains the path to a file or folder of structured report data to be parsed and then used to create a **Mapping File**. For convenience, a file or folder can be dragged and dropped on the **Structured Report(s)** field or a file or folder can be selected using the corresponding browse buttons.

Other features of the **Structure Report Extractor** are:

- **Mapping File** tab – displays a table of mapping values, which map the **Structured Report Search Term** (the complete branch name of an element) with a **Custom Mapping Field** (a simplified branch name with all spaces replaced by underscore characters) and **Rounding Precision** value (used to round off report values to a number of significant digits of accuracy). To identify potential problems in a mapping file, when it is loaded and duplicate **Structured Report Search Terms** are identified, the duplicate values are indicated with a warning icon.

- **Extraction Summary** tab – displays summary output from the structured report parser during an extraction operation.

- **Auto Text Dictionary** tab – displays an example Auto Text Dictionary that can be used for reference or as a starting point for creating or troubleshooting application use of mapping names and their structured report data values. Optionally, the Auto Text Dictionary can present a second column of information that shows which **Mapping Search Term** (from the **Mapping File**) corresponds to each **Mapping Result**.

- **Load** – when clicked, the currently selected **Mapping File** to be loaded into the **Mapping File** tab. Any mapping data loaded will be merged into any data already present on the **Mapping File** tab. The **Load** button also has a **Clear & Load** choice that first clears all data from the **Mapping File** tab before loading new data.
• **Apply** - when clicked, the currently loaded **Mapping File** is applied to the **Structured Report(s)**, with the results displayed on the **Auto Text Dictionary** tab. The Apply button also has an **Extract** choice that initiates an extraction operation where the currently selected **Structured Report(s)** are parsed, and the results of parsing are added to the **Mapping File** tab.

Columns in the **Mapping File** tab can be sorted in ascending and descending order by clicking on the column heading. Sorting can be cleared on a tab with an ordered column by typing **Alt-C** while viewing that tab. To clear sorting on both tabs, select the **Tools > Clear Sorting** menu choice.

A context menu appears when any row in the **Mapping File** tab is right-clicked.

![Context Menu](image)

Choices in the context menu perform the following:

- **Add Row** - enables a new row to be added to the **Mapping File** tab with initial, default values of “new search term”, “new_search_term”, and “1”.
- **Reset Row** - restores the **Custom Mapping Field** value on the currently selected row to the default mapping value, which is a simple replacement of all spaces with underscore characters.
- **Delete Row** - deletes the currently selected row from the **Mapping File** tab.
- **Unsort** - removes any sorting of columns that was done, displaying the **Mapping File** tab values in their original order.
- **Clear** - deletes all rows in the **Mapping File** tab, although it does not affect the **Mapping File** stored as a file, which is only changed by saving a new **Mapping File** using the same file name.
- **Show in Folder** - opens the containing folder of the current file.
- **Save** - stores the currently viewed **Mapping File** data as a text file and creates a backup of the previous file if it has the same name. **Save** can also be performed from the File menu.
- **Save and View** - performs a **Save** of the **Mapping File** data and then opens the file for viewing in the default system text file viewing application (e.g., Notepad).
- **Save As PDF** - saves all structured reports contained in the file or directory indicated by the **Structured Report(s)** field as a single PDF file and then opens the file for viewing in the default system PDF file viewing application (e.g., Acrobat). **Save As PDF** can also be performed from the File menu.
The **Options Dialog** in the **Structured Report Extractor** allows configuration of settings related to mapping file backups and extraction behavior, in addition to common logging settings.

- **Automatically create backup of previous mapping file** – causes an automatic backup of the previous mapping file of the same name to be created, incorporating a current timestamp in the filename.
- **Preserve existing Custom Mapping Fields on Extract** – when the **Extract** button is clicked, this option preserves existing modifications to Custom Mapping Fields in the Mapping File list. When unchecked, modifications can be overwritten.
- **Show Mapping File matches in Auto Text Dictionary** – when the **Apply** button is clicked, this option causes an extra column of information to be displayed in the **Auto Text Dictionary** showing which **Structured Report Search Term** field from the Mapping File matched to produce each **Mapping Result**. In the case where no search term matched, the value “nothing matched” is shown.
- **Highlight negations in Mapping File Search Terms** – when enabled, when negation terms are included in a mapping file search term field they will highlighted.
- **Flag Custom Mapping Fields longer than N characters** – when enabled, any Custom Mapping Field values in the Mapping File will be flagged to indicate values longer than the configured maximum valued (default is 40).
- **Keep duplicate SR Items** – when enabled, duplicate items (field names along with their values) will be kept. When more than one item with the same field name is found, the following naming scheme is used. Each of the N duplicate items will be numerically named in the order read (e.g., **FieldName_1, FieldName_2, ... FieldName_N**), with **FieldName** and **FieldName_N** containing the same item value and units. A special item **FieldName_0** will contain the number of duplicates (N) as its value.
Search functionality is provided by a **Find Dialog** customized for each tab. The **Mapping File** tab Find Dialog has options for case and whole word matches, search direction and use of wildcards.

The **Find All** button will search and highlight all rows that match the current search term. A convenient menu choice in the **Tools** menu will **Clear Find Highlighting** when selected.

The **Find** Dialogs on the **Extraction Summary** and **Auto Text Dictionary** tabs provide matching of the search term, with optional constraints of matching the case of the search term and matching whole words.

For easier viewing of the sometimes extensive contents of the **Extraction Summary** and **Auto Text Dictionary** tabs, the view can be zoomed in and out by typing **Ctrl-+** and **Ctrl-−** on the keyboard. Pressing **Ctrl-Home** while viewing either the contents of either of these two tabs resets the zoom level to normal. To reset zooming on both tabs, select the **Tools > Reset Zoom** menu choice.
4.5. HL7 Applications

PowerTools provides two Health Level 7 (HL7) applications for sending and receiving HL7 messages. The tools are under continuing development. An overview of each is provided below.

4.5.1. HL7 Receiver

The HL7 Receiver is a server that manages communication with an HL7 Sender client, receiving messages and optionally replying with ACK responses. The HL7 Receiver is started and stopped (or paused) using the run button and the stop (or pause) button.

In addition to typical options for TLS/SSL secure server connections, the user interface provides the following configurable options and information display components:

- **Port** – the port on which the server will listen for incoming connections and HL7 messages.
- **Send ACK** – when checked, causes the application to send a general acknowledgement ACK message to the HL7 Sender when it has received a message.
- **Send AR** – when checked, causes the application to send an AR acknowledgement ACK message to the HL7 Sender when it has received a message.
- **Messages Received** - is a count of the number of messages that have been received.
- **Quiet/Normal/Verbose** - the quantity of message content logged to the message window can be selected. **Quiet** produces a message once every 5,000 messages, **Normal** produces a single line of information, and **Verbose** also logs the entire text of each message.
- **Clear** - when clicked, clears the message window.

4.5.2. HL7 Sender

The HL7 Sender is a client that communicates with an HL7 Receiver server, sending messages and receiving optional ACK responses. In addition to typical TLS/SSL options for secure client
connections, the user interface provides the following configurable options and information display components:

- **Message Template** – A user-editable area that contains the template to be used for sending an HL7 message. The Message Template supports both drag and drop of a template file and copy and paste of a template into the text box. In the Tools menu, there is support for loading an example message template, clearing the template window, and reading and writing of message template files.
- **Message Sent/Response** – A logging window that displays messages sent and responses received.
- **Messages Sent** – a count of the number of messages that have been sent.
- **Quiet/Normal/Verbose** - the quantity of message content logged to the message window can be selected. Quiet produces a message once every 5,000 messages, Normal produces a single line of information and Verbose also logs the entire text of each message.
- **Clear** - when clicked, clears the message window.

**Sender**
- **Messages** – a count of the number of times the message template should be used to send a message. It is the total number of messages to be sent.
- **Send Delay** – a delay in seconds to be inserted between each successive message send.

**Receiver**
- **Host** – host name of the HL7 Receiver server.
- **Port** – port number that the HL7 Receiver server is listening on.
- **Application Name** – configurable name of an application that is sending the HL7 messages.
- 63 -

- **Sending Facility** - configurable name of the facility sending the HL7 messages.

**Connection**
- **New Connection Per Message** - when checked, sends each message on a new connection.
- **Wait for ACK** - when checked, causes the application to wait to receive an acknowledgement ACK message from the HL7 Receiver before sending the next message.
- **ACK Timeout** - how long to wait in seconds for an ACK to be received before timing out and continuing with the next message.

**Macro Replacement Table**
This is a table of **Macro Replacement** mapping rules to support dynamic, send-time HL7 message customization. Each occurrence of an item in the **Macro Name** column will be replaced in the message, before it is sent, with the corresponding **Replace Value**. For more details on using **Macro Replacement**, see below.

The **Options Dialog** in the **HL7 Sender** includes diagnostic settings that enable control over whether HL7 message framing characters are sent. This feature can be helpful when diagnosing HL7 message delivery problems.

- **Enable customization of HL7 message control** – enables or disables HL7 Message Control behavior. When disabled, all HL7 message framing characters are sent normally.
- **Send standard framing header before HL7 message** – when unchecked, framing header characters will not be sent.
- **Send standard framing trailer after HL7 message** – when unchecked, framing trailer characters will not be sent.
- **Send segment separator between HL7 message** – when unchecked, segment separator characters will not be sent.
Detailed logging of HL7 message flow, framing and segment separator characters, and HL7 message data in hex format can be enabled by setting the Log Level to Verbose.

In addition to the Options dialog and Revert To Factory Settings choice, several convenient features also are available in the Tools menu.

- **Load Example Message Template** - loads one of a selection of several common or standard message templates into the text box.
- **Clear Message Template** - clears the contents of the message template text box.
- **Read Message Template from File** - enables browsing for and loading a message template from a file.
- **Save Message Template to File** - enables saving the current contents of the message template text box to a file.

**Using Macro Replacement**

Simple Replacement

Macro Replacement provides a simple, convenient and powerful mechanism for automating HL7 message formatting. To use Macro Replacement, define a Macro Name that will be embedded in the message template and a Replace Value that determine the replacement behavior of the Macro Name. The Replace Value can be a simple string or one of the pre-defined macro functions.

For simple string value replacement, first define the desired Macro Name and ReplaceValue.

Next, insert a macro name string in the message template in the form: `{macro name}`
Using Macro Functions
Macro functions are used to insert generated values at a specific point in an HL7 message. To use macro functions, define a **Macro Name** as before. Insert a macro name string in the message template at the point you want the replacement to occur.

For the **Replace Value**, use one of the following functions:

**TIMESTAMP()**
**TIMESTAMP(format)**

This macro function will generate a date and time stamp in a desired format. The **format** is defined using standard date string formatting. For reference, refer to the online MSDN [Custom Date and Time Format Strings](https://msdn.microsoft.com/en-us/library/aa170059.aspx) documentation. Leaving the format empty will generate a timestamp for the current date and time. For example, to insert the current month, day and year, define the **Replace Value** as:

```
Macro Name  | Replace Value
---|---
today | TIMESTAMP(MM-dd-yyyy)
```

**GUID()**

This macro function will generate a new GUID at the desired position. Define the **Replace Value** using only the function:

```
Macro Name  | Replace Value
---|---
myguid | GUID()
```

To use these two specific examples of a TIMESTAMP and GUID, add the corresponding macro name references in the desired spots in the message template:

```
MSH|~\&|RAD|CDIC||${today}||ORM^O01|${myguid}|P|2.3
```

**Embedded Macro Functions**
For convenience, macro functions described above can be embedded directly inside the message template using the same format as was used above for the **Replace Value**: `{$macro function}`

For example, embedding a TIMESTAMP and GUID in the first line of the message template looks like this:

```
MSH|~\&|RAD|CDIC||${TIMESTAMP(yyyyMMddHHmmss)}||ORM^O01|${GUID()}|P|2.3
```

Note that the syntax for the macro functions is the same as when they are used in the **Replace Value** field.
5. Advanced Filtering

5.1. Composer Filter Action

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

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

<table>
<thead>
<tr>
<th>Tag</th>
<th>Input Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010,0010</td>
<td>(.*</td>
</tr>
<tr>
<td>0010,0020</td>
<td>(.*</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Tag</th>
<th>Output Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0010,0010</td>
<td>${2.1}</td>
</tr>
<tr>
<td>0010,0020</td>
<td>${1.1}</td>
</tr>
</tbody>
</table>

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** – 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 must specify how to split the Accession Number.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Input Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0008,0050</td>
<td>(^{10})(.*)</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Output Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0080,0050</td>
<td>${1.1}</td>
</tr>
<tr>
<td>0040,1001</td>
<td>${1.2}</td>
</tr>
</tbody>
</table>
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** – Take parts of the Accession Number and parts of the Requested Procedure ID and “mix and match” them.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Input Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0008,0050</td>
<td>(^.{10})(.*)</td>
</tr>
<tr>
<td>0040,1001</td>
<td>(^.{6})(.{4})</td>
</tr>
</tbody>
</table>

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, the first one having 6 characters and the second one having the following 4 characters. Then the outputs might look like this:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Output Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>0080,0050</td>
<td>${1.1}---${2.2}---${2.1}</td>
</tr>
<tr>
<td>0040,1001</td>
<td>${2.1}${1.2}</td>
</tr>
</tbody>
</table>

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 combined with plain text.

**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.

5.2. 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.
5.2.1. **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 5.2.5 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.
5.2.2. **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.
5.2.3. **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 \((x_1, y_1)\) and \((x_2, y_2)\) may be given as a percentage of the image’s rows \((y)\) and columns \((y)\). This option is the most flexible, as it does not require prior knowledge of the image size of the DICOM image to anonymize.

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

<table>
<thead>
<tr>
<th>Image Size: 800w (columns) x 600h (rows)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region 1: (x_1, y_1 = (0%, 0%)) (x_2, y_2 = (100%, 10%))</td>
</tr>
<tr>
<td>(x_1, y_1 = (0% \times 800 = 0, 0% \times 600 = 0))</td>
</tr>
<tr>
<td>(x_2, y_2 = (100% \times 800 = 800, 10% \times 600 = 60))</td>
</tr>
<tr>
<td>Region 2: (x_1, y_1 = (0%, -10%)) (x_2, y_2 = (100%, 100%))</td>
</tr>
<tr>
<td>(x_1, y_1 = (0% \times 800 = 0, -10% \times 600 = 540))</td>
</tr>
<tr>
<td>(x_2, y_2 = (100% \times 800 = 800, 100% \times 600 = 600))</td>
</tr>
</tbody>
</table>

**Negative Pixel Region Coordinates**

Note how the above region 2 was calculated with a \(y_1\) 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 as 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](image)

**Absolute Coordinates**

Both coordinates \((x_1, y_1)\) and \((x_2, y_2)\) 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 \((x_1, y_1)\) and \((x_2, y_2)\) 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](image)

**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](image)

The supported photometric interpretations for de-identification are:
- 74 -

- 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.

5.2.4. 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.DCS.DCSException: Failed to apply filter action --->

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
5.2.5. **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:

<table>
<thead>
<tr>
<th>Date Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study Date</td>
<td>2020/10/14</td>
</tr>
<tr>
<td>Content Date</td>
<td>2020/10/20</td>
</tr>
<tr>
<td>Origin Date</td>
<td>2046/08/08</td>
</tr>
<tr>
<td>Public Date Tag</td>
<td>0008,0020 – Study Date</td>
</tr>
</tbody>
</table>

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 demonstrates this in further detail.

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

<table>
<thead>
<tr>
<th>Study Date: 2020/10/14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Date: 2020/10/20</td>
</tr>
<tr>
<td>Origin Date: 2054/11/16</td>
</tr>
<tr>
<td>Calendar Date: 1986/02/22</td>
</tr>
</tbody>
</table>

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.

5.2.6. 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.

5.3. 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 **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).

### 5.3.1. Additional Notes

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)
6. Console Applications

6.1. Overview

The console applications contained in PowerTools are accessible via the PowerTools Console shortcut placed in the Start Menu on installation (see Section 2) and also from the shortcut to the PowerTools Console on the PowerTools Launcher GUI application. Many of the console applications are combined with the GUI version of the same application and are run using a -nogui flag.

All console (and GUI) applications contain a usage statement which can be displayed by running the application with a -h flag, although the -nogui flag is also required for those that are combined GUI and console applications. Note that most of the console applications are similar in functionality to their GUI counterparts.

For example, the GUI help message for the GUI version of the Query Client can be obtained by running this command:

```
DicomFilter -h
```

Running this command results in a help dialog popping up that looks like this:

![DICOM Filter Usage](image)

To view usage for the console version of the DICOM Filter, simply append the -nogui switch:

```
DicomFilter -nogui -h
```

The resulting output looks like this:

```
Filters one or more DICOM files on disk.
Files can be referenced either by an explicit path or by referencing a directory (which will be recursively searched for all DICOM files). Filtered DICOM files will be written back to their original location so it is recommended to make a backup of all files before filtering.

Usage: DicomFilter.exe FILTER_CFG_FILE -nogui [-1 LOG_LEVEL] -i PATH PATHS ... 
-h display command line usage
-nogui run the application in console mode
-1 LOG_LEVEL: if present, logging will be enabled and all log messages will be logged to the console using the LOG_LEVEL (1=[minimal], 2=[summary], 3=[detailed], 4=[verbose]) [default - summary] (PATH PATHS ...): a list of paths separated by spaces referencing either DICOM files or directories containing DICOM files to recursively filter using the FILTER_CFG_FILE filter configuration

GUI Mode: To display help for using this application in GUI mode, type: DicomFilter.exe -h
```
The detailed usage statement for each application provides user guidance for most tasks. Descriptions of some of the more common, complex uses are provided here for selected applications.

Note that any console application that supports DICOM has DICOM-specific logging support available using the -l command line switch. Options for logging level are: limited, summary, detailed, and verbose. To confirm logging availability, view the command line help (-h).

6.2. **DICOM to PDF**

**DICOM to PDF** is a console-only application that extracts Encapsulated PDF documents from one or more DICOM files and saves each PDF as a file. Note that the GUI version of the **Dicom File Editor** has Encapsulated PDF viewing and extraction support, as well.

6.3. **Indexer**

The **Indexer** is a console-only application that iterates over DICOM image files found in one or more file or directory paths and builds a CSV index file with information about those DICOM files.

6.4. **Modality Simulator**

**Modality Simulator** provides an advanced command-line interface to a simulated Modality Worklist Client (MWL SCU) which sends C-Find requests to a Modality Worklist Server (MWL SCP). The C-Find responses are used to build an HL7 ORM message that is sent to an HL7 Receiver to update the status. Note that this application requires a **Conductor** feature key to be present in the license. Usage information can be viewed by running the application in the console with the -h flag.

6.5. **MWL Client**

The **MWL Client** provides a command-line interface to common functionality of a modality worklist client suitable for testing, troubleshooting and as a developer tool. It is a console-only Modality Worklist Client (MWL SCU) application that queries a Modality Worklist Server. It supports standard and SSL/TLS communications.

Using the command-line interface, query parameters can be specified either directly using the -q flag, or via a configuration file. For example, a command line query for all patients that have the word “Simpson” in their name and have a Patient ID of “112233”, and returning those values plus Accession Number and Referring Physician, looks like this:

```
MwlClient [other params] -q 0008,0050= -q 0008,0090= -q 0010,0010=*Simpson*
```

Instructions for using a configuration file to perform queries are contained in the example configuration file (**MwlClientExample.cfg**) found in the PowerTools installation directory. Note that the GUI version of the **Query Client** supports top-level Modality Worklist queries, with future enhancements planned.
6.6. **MWL Server**

The **MWL Server** is a console-only Modality Worklist Server (MWL SCP) application that supports standard and SSL/TLS communications, and is distributed with a small worklist database of files. The following is a quick start to exploring the use of MWL Queries, such as by using the GUI version of the Query Client or the console MWL Client. To use the included worklist database, contained in the directory named *worklist* that is found inside the PowerTools installation directory, open a PowerTools Console from either the PowerTools Launcher or the Windows Start menu, and start the MWL Server by typing this:

```
MWLServer 11113 worklist
```

To query this worklist, start the Query Client application, enter a **Server Port** to match the port in the command above (11113), select a **Type & Method** of MWL, and with the **Use Default Tags** drop-down button select **Minimal MWL Tags**.

Then, click the **MWL Query** button and observe the output.

6.7. **QIDO Client**

The **QIDO Client** provides DICOMweb RS protocol support for sending queries to a QIDO-RS origin server. Examples of typical usage that contains the page URL, route, and a query string containing one or more tag queries, are:
Tag queries can be specified using either a keyword (e.g., PatientID) or a hex value (e.g., 00100010) and must be followed by an equal sign and optional match value to be correctly formatted. The supported tags vary by the particular QIDO server being queried. See below (Section 6.8) for the list of attribute tags supported by the PowerTools QIDO Server application.

When querying for multiple tags, use standard HTTP formatting for the query string, separating individual tag queries with an ampersand:

```bash
QidoClient -v -u http://localhost:11111/studies?PatientID=P01234&StudyDate=20210921&00100010=ABC*
```

The expected response format that is returned to the QIDO Client defaults to JSON in the absence of one of these optional flags (not shown in syntax example above):

- `-J` expect JSON responses
- `-X` expect DICOM+XML responses
- `-M` expect multipart/related DICOM+XML responses
- `-x` expect XML responses

### 6.8. QIDO Server

The QIDO Server provides DICOMweb RS protocol support for receiving and responding to queries from a QIDO-RS user agent using a DICOM dataset folder similar to that used by the Query Server. An example of typical usage is:

```bash
QidoServer -p 11111 -v -i D:\DicomImages
```

Note that the QIDO Server is located in the `netcore` subdirectory of the main PowerTools installation directory. Use of the QIDO Server requires installation of a recent version of the Microsoft ASP.NET Core Runtime Hosting Bundle as described in Section 2.1.

HTTPS support is available using the `-cf` and `-cp` flags to designate a TLS certificate filename and password.

The DICOM attribute tags that the QIDO Server supports, using either the keyword name or hex value, for queries from the PowerTools QIDO Client or a third-party tool are:

<table>
<thead>
<tr>
<th><strong>Keyword Name</strong></th>
<th><strong>Hex Value</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessionNumber</td>
<td>00080050</td>
</tr>
<tr>
<td>AdditionalPatientHistory</td>
<td>001021B0</td>
</tr>
<tr>
<td>AdmittingDiagnosesDescription</td>
<td>00081080</td>
</tr>
<tr>
<td>EthnicGroup</td>
<td>00102160</td>
</tr>
<tr>
<td>InstanceNumber</td>
<td>00200013</td>
</tr>
<tr>
<td>IssuerOfPatientID</td>
<td>00100021</td>
</tr>
<tr>
<td>ModalitiesInStudy</td>
<td>00080061</td>
</tr>
</tbody>
</table>
6.9. Query Client (console version)

6.9.1. Query example

As described in the application’s command line help, the Query Client can make use of user-defined query configuration and input filters files. For example, if a user desired to query a server for datasets with a specific study date and filter one or more tags, the following steps can be used:

1. Create a query configuration file containing the tag or tags to be queried for. In this example, create a file called “query.txt” that contains the following lines:

   
   0008,0020 = 20070101
   0010,0010 =
   0010,0020 =
   0020,000d =

   This queries for datasets that have a Study Date (0008,0020) of January 1, 2007 and will return values of the Study Date, Patient’s Name (0010,0010), Patient ID (0010,0020) and Study Instance UID (0020,000d) tags.
Note that the Query/Retrieve level should NOT be put in the query.txt file, as it is set using a command line switch (default level is STUDY).

For a description of the query configuration file format, see below or view the example QueryClientExample.cfg file in the PowerTools installation directory.

2. Create a filter file using the GUI version of the Query Client application. A simple filter to create for this example would be one that modifies the Patient’s Name by changing its case to all lowercase. See below for a description of using the Filter Editor in one of these applications to create an XML filter file, saving it as “filter.xml”.

3. Run the Query Server console application, pointing it at a directory that contains at least one dataset that matches the query defined in “query.txt”, like this:

   QueryServer 11112 path_to_dataset_folder

4. Query the server using the console version of the Query Client, taking care to use the required -nogui switch, like this:

   QueryClient localhost 11112 -nogui -f query.txt -fi filter.xml

Be sure either to have the “query.txt” and “filter.xml” files in the same directory from which you are running the command, or to specify on the command the complete path to each file. The output will look something like this, noting that the Patient’s Name is now lowercase:

   Query Result 1:
   (0008, 0020) DA( 8) Study Date 20070101
   (0008, 0052) CS( 6) Query/Retrieve Level STUDY
   (0010, 0010) PN(12) Patient’s Name doe^john
   (0010, 0020) LO(14) Patient ID 000003F136E8A
   (0020, 000d) UI( 54) Study Instance UID 1.2.840.114089.1.0.1.3229814446.1265295723.2160.868861

6.9.2. Creating a Filter File

To create a filter file using Filter Editor in the GUI version of the Query Client application, run the Query Client, open the Options dialog from the Tools menu, and click on the Edit Filters button. The Filter Editor dialog will open.
Click the button to add a new filter. Then, in the **Conditions** area select “Dicom Tag”, and define it as “Patient’s Name” with a condition of “Exists”.

In the **Actions** area, select **Change Case** for “Patient’s Name” and define the action as “Upper”. The dialog should now look like this:

![Filter Editor Dialog](image)

Finally, click **Export**, and save this filter with a meaningful name, such as “filter.xml”. Close all dialogs and exit from the GUI version of the **Query Client**.

For convenience, this same **Filter Editor** is available in the GUI versions of the **DICOM Filter**, **DICOM File Editor**, **Store Client**, **Store Server**, **Verification Client**, and **Verification Server** applications. More details about using the **Filter Editor Dialog** are found in section 4.1.3 above.

### 6.9.3. Format of Query Configuration File

The query configuration file described above has a simple name-value pair format. For reference, there are several example configuration files (*.cfg) in the PowerTools installation directory.
Each line of the file contains a DICOM group and element number separated by a comma followed by an equals sign and an optional expected value. For example:

```
0008,0020 = 20070101
0010,0010 =
```

**Match line** - The first line is a “match line” and contains the group and element numbers for a Study Date tag. When used in a query, this line will match a Study Date tag with the specific date defined (January 1, 2007).

**Result line** - The second line is a “result line” and contains the group and element numbers for a Patient’s Name tag. Because no value is specified, this line will match any Patient’s Name.

**Results returned** - When all match lines in a query configuration file match those in a dataset on the Query Server, that dataset is deemed a match. The results returned by a query will consist of all tags and values that match any of the match lines or result lines. In the above example, if a dataset is found with the matching Study Date, the results returned will consist of the Study Date and Patient’s Name.

### 6.10. Query Server

The Query Server is a console-only Query/Retrieve Server that responds to C-Find and C-MOVE requests from a Query/Retrieve Client. The Query Server supports SSL/TLS communication, and provide on-demand directory traversal when a query includes any of the following, supported, calculated attributed tags:

- Number of Patient Related Studies
- Number of Patient Related Series
- Number of Patient Related Instances
- Number of Study Related Series
- Number of Study Related Instances
- Number of Series Related Instances
- Modalities in Study
- SOP Classes in Study

Note that the Alternate Representation Sequence tag is not yet supported.

### 6.11. RIS Simulator

RIS Simulator provides an advanced, command-line interface to an HL7 Sender which sends HL7 ORM messages to a Modality Worklist Server (MWL SCP). Note that this application requires a Conductor feature key to be present in the license. Usage information can be viewed by running the application in the console with the `-h` flag.
6.12. STOW Client

The STOW Client provides DICOMweb RS protocol support for storing to a STOW-RS origin server in one of three modes: Part 10, JSON or XML. Examples of typical usage are:

```
StowClient -mode PART10 -v -u http://localhost:11111/studies
    -f D:\Temp\Images\DICOM\mr-knee.dcm
    -f D:\Temp\Images\DICOM\mr-hip.dcm

StowClient -mode JSON -v -u http://localhost:11111/studies
    -pi 12345 -pn "Smith^John" -st 1.2.3.4.5 -sr 2.3.4.5.6
    -sc 1.2.840.10008.5.1.4.1.1.2 -si 3.4.5.6.7 -m CT
    -f D:\Temp\Images\JPEG\ct-skull.jpg
    -f D:\Temp\Images\JPEG\ct-ankle.jpg

StowClient -mode XML -v -u http://localhost:11111/studies
    -pi 12345 -pn "Smith^John" -st 1.2.3.4.5 -sr 2.3.4.5.6
    -sc 1.2.840.10008.5.1.4.1.1.2 -si 3.4.5.6.7 -m CT
    -f D:\Temp\Images\JPEG\ct-skull.jpg
    -f D:\Temp\Images\JPEG\ct-ankle.jpg
```

Viewing the usage message for each of these three modes requires including the `mode` argument in addition to the help flag:

```
StowClient -h -mode PART10
StowClient -h -mode JSON
StowClient -h -mode XML
```

The expected response format that is returned to the STOW Client in any of its three modes (Part 10, JSON, XML) defaults to JSON in the absence of one of these optional flags (not shown in syntax examples above):

```
-J expect JSON responses
-X expect DICOM+XML responses
-x expect XML responses
```

In each of these modes, up to 100 images can be sent. This limit is based on the maximum length of a Windows command-line string (8191 characters per Microsoft documentation) combined with estimates for how long each typical image file path might be.

When run in Part 10 mode, the STOW Client accepts an optional `batch size` argument (-b) that specifies how many images to include in each store request. The default `batch size` is 5.
6.13. **STOW Server**

The **STOW Server** provides DICOMweb RS protocol support for simulated storing from a STOW-RS user agent. Although studies received by the **STOW Server** are not stored, it produces simulated storage responses based on the protocol. An example of typical usage is:

```
StowServer -p 11111 -v
```

Optionally, a directory can be specified to store received Part 10 DICOM images using the `-i` switch. The default behavior is to not store received images.

```
StowServer -p 11111 -v -i D:\Images
```

Note that the **STOW Server** is in the `netcore` subdirectory of the main PowerTools installation directory. Use of the **STOW Server** requires installation of a recent version of the Microsoft ASP.NET Core Runtime Hosting Bundle as described in Section 2.1. HTTPS support is available using the `-cf` and `-cp` flags to designate a TLS certificate filename and password.

6.14. **WADO Client**

The **WADO Client** provides DICOMweb RS protocol support for sending queries to a WADO-RS server. An example of typical usage is:

```
WadoClient -v -u http://localhost:11111/studies/1.2.3.4.5
```

Optionally, a directory can be specified to store received Part 10 DICOM images using the `-i` switch. The default behavior is to not store received images.

```
WadoClient -u http://localhost:11111/studies/1.2.3.4.5 -i D:\Images
```

Additional options are described in the command line usage. Note that the **WADO Client** is located in the `netcore` subdirectory of the main PowerTools installation directory. Use of the **WADO Server** requires installation of a recent version of the Microsoft ASP.NET Core Runtime Hosting Bundle as described in Section 2.1.

6.15. **WADO Server**

The **WADO Server** provides DICOMweb RS protocol support for receiving and responding to queries from a WADO-RS client using a DICOM dataset folder similar to that used by the **Query Server**. An example of typical usage is:

```
WadoServer -p 11111 -v -d D:\DicomImages
```

Note that the **WADO Server** is in the `netcore` subdirectory of the main PowerTools installation directory. Use of the **WADO Server** requires installation of a recent version of the Microsoft ASP.NET Core Runtime Hosting Bundle as described in Section 2.1. HTTPS support is available using the `-cf` and `-cp` flags to designate a TLS certificate filename and password.
7. Overriding DICOM Framework Defaults

The underlying DICOM framework upon which the PowerTools applications that use DICOM are built is named the Laurel Bridge Software DICOM® Connectivity Framework (DCF). The DCF™ has a variety of default values, many of which can be manually overridden by creating and editing a settings configuration file in a very specific format. This capability is supported in any GUI or console application in PowerTools that makes use of DICOM and therefore the DCF.

It is recommended that caution should be exercised when overriding a default setting, as many of the settings can have a significant effect on the function and performance of PowerTools applications. Guidance on overriding default values may be sought by contacting Laurel Bridge Software for support.

The steps needed to override a default value in an application are as follows. This example assumes the application file is Example.exe and that pdu_read_timeout_seconds is the setting being overridden:

- Exit from Example.exe if it is running.

- Make a copy of the example_overrides.dcfcfg file found in the PowerTools installation directory. Be sure to not make changes to the original configuration file so that it can be referred to as needed in the future.

- Rename the copy to Example.dcfcfg, leaving it in the same installation directory where the Example.exe application is located.

- Edit Example.dcfcfg in a favorite text editor, finding the desired setting (in this case pdu_read_timeout_seconds), uncommenting it, and revising its value to the desired new value. The resulting line in the revised configuration file will look like this:

  pdu_read_timeout_seconds = 300

- Save Example.dcfcfg and run Example.exe. The overridden default(s) will be used.

Note that more detailed directions for overriding default DCF settings are contained in comments inside example_overrides.dcfcfg. These comments include notes about selected default values that are already being overridden in one or more of the PowerTools applications as well as a complete listing of the names, default values, and descriptions for available settings.
8. Application Summary

There are three forms of applications in PowerTools: combined GUI and console applications, GUI-only, and console-only. For clarity, they are listed here. Usage information for the GUI version of applications and console-only applications is displayed using the \texttt{-h} flag, while usage information for the console version of combined GUI and console applications is displayed using both the \texttt{-h} and \texttt{-nogui} flags.

8.1. Combined GUI and Console Applications

- DICOMDIR Creator
- DICOM File Editor
- DICOM Filter
- Dictionary Finder
- Query Client
- Store Client
- Store Server
- Study Generator
- Verification Client
- Verification Server

8.2. GUI-only Applications

- PDF to DICOM
- HL7 Receiver
- HL7 Sender
- Launcher
- Structured Report Extractor

8.3. Console-only Applications

- DICOM to PDF
- Indexer
- Modality Simulator
- MWL Client
- MWL Server
- QIDO Client
- QIDO Server
- Query Server
- RIS Simulator
- STOW Client
- STOW Server
- WADO Client
- WADO Server
## 9. PowerTools Quick Reference

### General

<table>
<thead>
<tr>
<th>GUI Usage</th>
<th>Console Usage</th>
<th>Rename app window</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appname.exe -h</td>
<td>Appname.exe -h -nogui</td>
<td>Appname.exe -name &quot;new name&quot; See example_gui_renaming.bat</td>
</tr>
</tbody>
</table>

### Applications

<table>
<thead>
<tr>
<th>Launch any PowerTools app</th>
<th>View, edit, modify DICOM files</th>
<th>Lookup SOP Class &amp; Transfer Syntax UIDs, element names, tags, VR types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launcher</td>
<td>Dicom File Editor</td>
<td>Dictionary Finder</td>
</tr>
<tr>
<td>Create DICOMDIR from images</td>
<td>Filter DICOM files, create filters</td>
<td>Extract Structure Report data, generate mapping file for integrated dictation systems (e.g. PS360)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Send HL7 messages</th>
<th>Run console apps</th>
<th>Send DICOM dataset to server</th>
<th>Create encapsulated PDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL7 Sender</td>
<td>PowerTools Console</td>
<td>Store Client</td>
<td>PDF to DICOM</td>
</tr>
<tr>
<td>Receive HL7 messages</td>
<td>Query a PACS (Find, Move, MWL)</td>
<td>Receive DICOM datasets, store to disk</td>
<td>Generate test datasets, send to DICOM server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Send echo/ping to DICOM device</th>
<th>Autorun on GUI app start</th>
<th>Override default DICOM settings</th>
<th>Load DICONDE or ext dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification Client</td>
<td>Store Server &amp; Study Generator, -run switch</td>
<td>Copy and edit example_overrides.dcf</td>
<td>Launcher, click upper-left DICOM icon</td>
</tr>
<tr>
<td>Receive &amp; respond to C-Echo ping messages</td>
<td>Test DICOMweb QIDO-RS communication</td>
<td>Test DICOMweb STOW-RS communication</td>
<td>Test DICOMweb WADO-RS communication</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Load DICONDE or ext dictionary</th>
<th>Load DICONDE or ext dictionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launcher, click upper-left DICOM icon</td>
<td>Launcher, click upper-left DICOM icon</td>
</tr>
<tr>
<td><strong>Dicom File Editor</strong></td>
<td><strong>Study Generator</strong></td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td><strong>View file or directory of images</strong></td>
<td><strong>Store studies in parallel</strong></td>
</tr>
<tr>
<td>Drag folder onto Dicom File Editor</td>
<td>Check Store studies in parallel</td>
</tr>
<tr>
<td>Use Open or Open Folder menu option</td>
<td>Use separate association</td>
</tr>
<tr>
<td><strong>Compare elements in images</strong></td>
<td><strong>Demo multiple study generators</strong></td>
</tr>
<tr>
<td>Select element in current image, select a different image in hierarchy list</td>
<td>Run example_gui_renaming.bat</td>
</tr>
<tr>
<td><strong>Save DICOM image as plain image</strong></td>
<td><strong>Quickly change target store server</strong></td>
</tr>
<tr>
<td>Use Save Image As menu option (bmp, gif, jpeg, pbm, pgm, png, ppm, tiff)</td>
<td>Use Store Server Profiles, define multiple, easily select desired server</td>
</tr>
<tr>
<td><strong>Copy image path to clipboard</strong></td>
<td><strong>Delay between stores</strong></td>
</tr>
<tr>
<td>Right-click on Header, Hex or Image tab, or image panel, select Copy File Path to Clipboard</td>
<td>Change delay in Options dialog</td>
</tr>
<tr>
<td><strong>Save image shortcut</strong></td>
<td><strong>Set background image</strong></td>
</tr>
<tr>
<td>Right-click on a tab or image panel, select Save Image As</td>
<td>Click on Background Image or Browse button</td>
</tr>
<tr>
<td><strong>Open containing directory</strong></td>
<td><strong>Show tag overlay</strong></td>
</tr>
<tr>
<td>Right-click on a tab or image panel, select Show in Folder</td>
<td>Check Tag Overlay box</td>
</tr>
<tr>
<td><strong>Load DICONDE or ext dictionary</strong></td>
<td><strong>Set frame count and image dimensions</strong></td>
</tr>
<tr>
<td>Click upper-right DICOM indicator (if present) or Manage button on Options dialog</td>
<td>Set Frames/Image, Columns and Rows values</td>
</tr>
<tr>
<td><strong>Show names for well-known UIDs</strong></td>
<td><strong>Max frame size</strong></td>
</tr>
<tr>
<td>Check Show names for well-known UIDs on Options dialog</td>
<td>400 MB (381 MiB)</td>
</tr>
<tr>
<td><strong>Save DICOM image as plain image</strong></td>
<td><strong>Use separate association</strong></td>
</tr>
<tr>
<td><strong>Demo image overlays</strong></td>
<td>Select Study, Series or Image</td>
</tr>
<tr>
<td><strong>Add example tags</strong></td>
<td><strong>Send images in various sizes, view in Store Server</strong></td>
</tr>
<tr>
<td><strong>Max number of images generated</strong></td>
<td></td>
</tr>
<tr>
<td>2^31 - 1</td>
<td>2,147,483,647</td>
</tr>
<tr>
<td><strong>Clear background image</strong></td>
<td><strong>Use “SAMPLE” watermark</strong></td>
</tr>
<tr>
<td>Click Clear button</td>
<td>Shown only when Background Image is used</td>
</tr>
<tr>
<td><strong>Set frame count and image dimensions</strong></td>
<td><strong>Override bits allocated &amp; stored (see tooltip)</strong></td>
</tr>
<tr>
<td>Set Frames/Image, Columns and Rows values</td>
<td>Add “Bits Allocated and Stored” tags in Tools menu, modify values</td>
</tr>
<tr>
<td><strong>Max frame size</strong></td>
<td><strong>Change Transfer Mode</strong></td>
</tr>
<tr>
<td>400 MB (381 MiB)</td>
<td></td>
</tr>
</tbody>
</table>
10. **Conductor**

10.1. **Overview of Conductor**

Laurel Bridge **Conductor** is an application controller and performance measurement tool in the Laurel Bridge PowerTools™ suite. Its primary uses are for performance testing proposed scenarios on a virtual network as a means of proof-of-concept or predictive analysis. An add-on feature key must be purchased to enable Conductor functionality in PowerTools.

This user manual covers general installation, setup and licensing of Conductor on a single machine or virtual network, creation of and working with scenarios, and running Conductor from the command line to perform performance regression tests.

Conductor consists of a **Controller Application** and an **Agent Service**. The controller provides an interface that enables the user to create, manage and control scenarios that involve one or more applications, and to gather and view runtime performance data from those applications. The agent service runs on all systems involved in a scenario and is responsible for marshalling commands and data to and from the Conductor controller and the applications being controlled. This marshalling is done using embedded web server technology that accepts commands and data via standard HTTP GET and POST requests and responses.

The first envisioned use of Conductor is to measure performance and throughput of proposed network system configurations, measure the impact on throughput of DICOM images sent across the network under a variety of circumstances, such as via different Compass modes or with some amount of HL7 traffic competing for resources. Many other uses are possible, limited only by available resources and the capability to report and gather the desired runtime information.

Because Conductor is a powerful and complex tool, it is recommended for use by technically knowledgeable users who are familiar with the Laurel Bridge Compass™ product and with PowerTools software generally. Specific beneficial background includes facility with managing and configuring a virtual network and working with PowerTools applications.
10.2. Installing Conductor

Conductor is included by default in PowerTools, beginning with version 1.0.43. To install PowerTools, run the installer executable (Setup.exe) and follow the on-screen directions. For reference, see the Installation section of the PowerTools User Manual.

10.2.1. Single System

Installing on a single system is straightforward. Install PowerTools. Optionally, install Compass if you plan to run it locally as part of a test scenario. Follow steps below for starting the agent service and then for configuring and running a scenario using Conductor.

10.2.2. Virtual Network

Installation on a virtual network is a more involved process. These instructions are based on experience with a specific virtual network with every attempt made to write them in a generally applicable way. More details that expand on the general descriptions here are found further below in this user manual. Steps for setting up Conductor on a virtual network are:

1. Acquire PowerTools and Compass installers and necessary license files for running them on your virtual network. The PowerTools license on the server on which you will run Conductor must have the conductor feature key enabled, while other installations of PowerTools do not require that feature key at present.
2. Move the PowerTools and Compass installers into a known location on a gateway server on your virtual network, along with PowerTools and Compass license files. A recommended approach is to have a single gateway machine that is the only machine reachable from outside the virtual network. Make sure that the directory containing these installers is available from all other servers on the virtual network.
3. On each server on the virtual network, install PowerTools and a corresponding license. Also, on each server start the Conductor Agent Service.
4. On the gateway server (recommended), install the PowerTools license that has the conductor feature key enabled. You will run Conductor on that server so that you have the option to view browser-based output in a browser outside of the virtual network.
5. On the server(s) on which you plan to run Compass, install Compass and the corresponding license. Note that you do not need to manually start the Compass service, as it will be started the first time you open the Compass user interface.
6. On the same Compass server(s), verify that Receive Side Scaling (RSS) is Enabled and Jumbo Packet is set to Jumbo 9000, as performance otherwise can be impacted. To do this, find and open the operating system View Network Connections panel, right-click on the relevant connection (e.g., “Ethernet 2”), click Configure and go to the Advanced tab. Scroll down to Jumbo Packet and set its value to Jumbo 9000, then scroll further down to Receive Side Scaling and set its value to Enabled, and finally click OK. The setting will be applied immediately.

Once these setup steps are complete, you are ready to configure a Conductor scenario and run it.
10.3. Getting Started with Conductor

10.3.1. Installing a License

In order to make use of Conductor, a PowerTools license that includes the Conductor feature key must be installed only on the system on which the Conductor controller will run. It is not necessary on any of the agent systems. This license must be specially acquired from Laurel Bridge Software and saved as a local file. Instructions for installing a license are included in the PowerTools User Manual, though the license can be installed directly using Conductor.

To install directly, run Conductor manually by navigating to the PowerTools installation directory (C:\Program Files\Laurel Bridge Software\PowerTools) and double-clicking on the Conductor icon. Because Conductor is still considered an engineering tool rather than a developer tool, it is not yet included in the Windows start menu or the PowerTools Launcher interface. After the application appears, select Help > About… menu option. When the About Conductor dialog appears, click the Install License… button, browse to the license file, and hit OK. Information on the status of the license, including when it expires, can be found on this dialog available at any time by selecting Help > About….

A typical setup for Conductor is within a sequestered virtual network, so activation of the license may not be possible over the Internet. However, if access is available, helpful instructions for Activating a License over the Internet are found in the PowerTools User Manual. Additional assistance is available by sending email to: support@laurelbridge.com

10.3.2. Starting the Conductor Agent Service

To enable the Conductor Controller to issue commands and gather runtime data from other PowerTools applications, the Conductor Agent service that is installed when PowerTools is installed must be started.

The Conductor Agent Service can be started in a variety of ways, including:

- Open a command prompt, running as Administrator, and type the following command:
  
  net start conductor
• Open the Windows Task Manager, navigate to the Services tab, scroll down and right-click on the Conductor (Conductor Agent Service) entry, and select **Start**.
• From the Windows Control Panel, open the Administrative Tools, then open the Services interface. Right-click on the Conductor Agent Service entry and select **Start**.

To verify details about the Conductor Agent Service and whether it is running, view its status in the Services interface you used to start it or run these command:

```
sc query conductor
sc qc conductor
```

10.3.3. **Starting the Conductor Controller**

The **Conductor Controller**, simply called **Conductor**, is run by locating and double-clicking its program icon.

10.3.4. **Conductor User Interface**

The Conductor user interface consists of a menu, general scenario information area, a start and stop button, four application configuration tabs, a status logging window, and a status bar that displays running time information.
**Scenario Information**

To use Conductor, a *scenario* is configured and then run. A *scenario* is a collection of configuration settings that includes a scenario name, a runtime duration, a reporting period, and specific settings for one or more Compass, Store Server, Study Generation, HL7 Receiver and HL7 Sender applications from the PowerTools suite. Each of these applications has a variety of settings available that can be configured and stored with a scenario.

The scenario information at the top of the Conductor interface displays the name of the currently loaded scenario (or “modified scenario” if any changes have been made to the current scenario since it was loaded or saved, or empty scenario” if there is no scenario loaded or configured).

The **Run for** text box is where the desired duration of a scenario run can be set. A scenario can be configured to run for a short as 1 minute up to 99 hours, 59 minutes (4 days, 3 hours, 59 minutes). The **Report every** text box is where the desired stats report period is configured which determines how often all of the running applications in the scenario report their recently gathered runtime stats back to Conductor. This report period can be set to as short as 10 seconds or as long as 99 minutes, 59 seconds.

The **Start** and **Stop** buttons enable a scenario to be manually started and stopped.

**Menu**

The **File** menu contains the following selections:

- **Load Scenario** - select and load a scenario.
- **Save Scenario** - save the current scenario configuration to a file.
- **Clear Scenario** - clear the current scenario, resetting Conductor to an empty scenario.
- **Save Summary to CSV** - save a higher level summary of the runtime status of a scenario to a CSV file.
- **Save Report to HTML** - save the HTML Report for the current scenario (see **View Report in Browser**) to an HTML file.
- **Save Stats Log to CSV** - save all gathered stats from the most recent or ongoing run of a scenario to a CSV file for further analysis as desired.
- **Exit** - exit Conductor, first stopping the scenario if it is running and offering the user a change to save the gathered log of stats before exiting.

The **View** menu contains the following selections:

- **View Agent Status** - view the agent status of all servers and Compass applications that are active parts of the current scenario. The status dialog may take a few seconds to appear if one or more of the agents are not alive, since it waits for a short time before reporting that an agent service is not alive.
- **View Agent Diagnostics** - view a dialog that displays an explanation and two links that can be used to view any agent service diagnostic interface and to verify that the Conductor built-in web server is reachable.
- View Summary - displays a quick view summary in a text-only dialog box of the currently running scenario, with information about the applications, run status and throughput of the scenario.
- View Report in Browser - displays the same summary information as the quick view summary along with graphs of a variety of runtime statistics gathered from any of the active Compass, Waypoint, Store Server, Study Generator, HL7 Receiver, HL7 Sender, and Command Line applications in the current scenario.
- View Stats Log in Browser - view the currently gathered runtime stats in a web browser in an easily viewed format.
- View Saved Report in Browser - select and view one of the auto-saved HTML Reports in a web browser.
- View Saved Stats Log in Browser - select and view one of the auto-saved Stats Log CSV files in a web browser.
- Show Backup Stats Logs & Reports Folder - opens the directory where Conductor periodically saves an automatic copy of the gathered runtime stats and HTML reports. This directory (Backup) is adjacent to the Conductor application Configuration and Log directories, with default locations inside the user’s AppData\Roaming\Laurel Bridge Software\PowerTools\Conductor directory.

The Tools menu contains the following selections:
- Use Verbose Status Logging - enabling verbose status logging causes more detailed and frequent messages to be displayed in the Status log area of the user interface. This option can be enabled and disabled while a scenario is running, if desired.
- Backup Stats Log & Report Every - configures how frequently the current gathered runtime stats are written to a backup CSV file and the corresponding HTML Report in the Backup directory. Choices are every 1, 5, 10, 30 or 60 minutes.
- Number of Backup Stats Logs & Reports to Keep - configures the maximum number of Stats Logs and HTML Reports to keep in the Backup directory. Choices are every 1, 5, 10, 25 or 100.
- Set PowerTools Application Path - enables the default application path, the PowerTools installation directory, to be overridden. This can be helpful if a custom or development version of PowerTools is installed for Conductor testing purposes.
- Reset PowerTools Path to Default - resets the PowerTools Application Path to its default value: C:\Program Files\Laurel Bridge Software\PowerTools
- Clear Compass with Scenario - if checked, it enables the Clear Compass Database & Images command to be issued when a scenario is started.
- Clear Compass Database & Images - sends a command to the agent running on each active Compass system in the current scenario requesting that the Compass database and image directory be cleared. A message with the result of the request is displayed in the Status log.
- Clear Compass with Scenario - if checked, it enables the Clear Waypoint Database command to be issued when a scenario is started.
- Clear Waypoint Database - sends a command to the agent running on each active Waypoint system in the current scenario requesting that the Waypoint database be cleared. A message with the result of the request is displayed in the Status log.
• **Try to Start Agents with Scenario** - if checked, an attempt will be made to start any stopped agents *(Start All Agents)* before then starting the scenario.

• **Start All Agents** - sends a system level request to each agent system, requesting that its Conductor Agent Service be started if it is not already running. Due to operating system level permissions, it is possible that this command will not work, though it often does.

• **Stop All Agents** - sends a system level request to each agent system, requesting that its Conductor Agent Service be stopped if it is running. Due to operating system level permissions, it is possible that this command will not work, though it often does.

• **Nuke All Agents** - this powerful option sends a command to each agent in the current scenario, requesting that all running applications be immediately terminated. Though rarely needed, this feature can be handy in complex testing environments of many servers running many applications where an occasional application refuses to exit for no discernable reason.

• **Revert to Factory Settings** - restores *Conductor* to its initial configuration including an empty scenario.

The Help menu contains the following selection:


• **About** - displays the *About Conductor* dialog that includes the version number of the installed PowerTools, copyright and license information, and a status message about whether a more recently released version of PowerTools is available (if Internet connectivity is available to make this check).

![Agent Status dialog showing “not alive” and “alive” statuses.](image)

![Agent Diagnostics dialog.](image)
Example output in the Summary Quick View dialog. Similar summary information is displayed in the web-browser Report view.

**Application Configuration Tabs**

There are four tabs used for configuration Compass & Waypoint, DICOM, HL7 and Command Line applications.

Available configuration options for Compasses are:

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
- **Server** - the name or IP address of the server on which Compass is running.
- **App Config** - allows an optional Compass application configuration file to be copied to the correct location on the Server so that Compass will use it. Use this option when you would like to try different configurations of Compass (e.g., comparing Store & Forward to Direct modes) without having to manually reconfigure Compass. Note that these Compass config files will need to be created manually using Compass and then saved to a file for use by Conductor.
• **Exe Config** - less commonly needed, this allows an optional Compass execution configuration file to be copied to the correct location the **Server** so that Compass will use it.
• **Description** - an optional comment or explanation added by the user for reference.

The **App Config** and **Exe Config** options can be cleared by right-clicking on either and selecting **Remove Config File**.

Available configuration options for **Waypoints** are:

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
- **Server** - the name or IP address of the server on which Compass is running.
- **App Config** - allows an optional Waypoint application configuration file to be copied to the correct location on the **Server** so that Waypoint will use it. Use this option when you would like to try different configurations of Waypoint without having to manually reconfigure Waypoint. Note that these Waypoint config files will need to be created manually using Waypoint and then saved to a file for use by Conductor. The **App Config** option can be cleared by right-clicking on it and selecting **Remove Config File**.
- **Description** - an optional comment or explanation added by the user for reference.

Available configuration options for **Store Servers** are:

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
- **How Many** - enter how many of this application should be run in the current scenario. Maximum is 99.
- **On Server** - the name or IP address of the server on which the application should be run.
- **First Port** - indicates the port on which the first Store Server running on the specified server will listen on. If more than one Store Server is to be run, subsequent Store Servers
will listen on subsequent ports. If the specific **First Port** is 11115 and four Store Servers are to be run, they will listen on ports 11115, 11116, 11117 and 11118, respectively.

- **Description** - an optional comment or explanation added by the user for reference.

Available configuration options for **Study Generators** are:

<table>
<thead>
<tr>
<th>Run</th>
<th>How Many</th>
<th>On Server</th>
<th>Host Server</th>
<th>First Host Port</th>
<th>Port Mode</th>
<th>Transfer Mode</th>
<th>Studies</th>
<th>Series</th>
<th>Images</th>
<th>Frames</th>
<th>Frame Size</th>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>8</td>
<td>localhost</td>
<td>localhost</td>
<td>11112</td>
<td>Same</td>
<td>DIMSE</td>
<td>10</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>512</td>
<td>settings</td>
<td></td>
</tr>
</tbody>
</table>

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
- **How Many** - enter how many of this application should be run in the current scenario. Maximum is 99.
- **On Server** - the name or IP address of the server on which the application should be run.
- **Host Server** - the name or IP address of the server to which Study Generators should send the images they generate.
- **First Host Port** - indicates the port to which the first Study Generator running will send the images it generates. If more than one Study Generator is to be run, subsequent Study Generators will use the **Port Mode** option to either send images to the same port or to sequential ports.
- **Port Mode** - the port sequencing mode that determines port numbers used by Study Generators. If the **Port Mode is Same**, all Study Generators will send to the same port indicated by **First Host Port**. If the **Port Mode is Sequential**, Study Generators will send to sequential port numbers on the **Host Server**, starting with the **First Host Port**.
- **Transfer Mode** - the mode used to send images to a server, which can be **DIMSE, STOW Part10, STOW JSON**, or **STOW XML**. Note that The **STOW JSON** and **STOW XML** modes are limited to generating single frame images. The selected mode determines which **Study Generator Settings** are configurable for a given application entry.
- **Studies** - total number of studies to generate in the scenario. Maximum is 1,000,000.
- **Series** - number of series to generate for each study. Maximum is 1,000,000.
- **Images** - number of images to generate for each series. Maximum is 1,000,000.
- **Frames** - number of frames to generate for each image. Maximum is 10,000. The **STOW JSON** and **STOW XML** modes are limited to a single frame, so if the value of Frames is larger than 1 the warning icon is displayed with a tooltip that explains that only a single frame will be generated and sent.
- **Frame Size** - size of each frame in KB. The resulting generated image is the number of Frames x the Frame Size. Maximum is 400,000 KB.
- **Settings** - enables configurations using the **Study Generator Settings** dialog (below).
- **Description** - an optional comment or explanation added by the user for reference.

The **Study Generator Settings** dialog enables several additional options to be configured. While most settings are self-explanatory, details on each are covered in the Study Generator section of the PowerTools user manual.
Of note are the following options on the *Study Generator Settings* dialog:

- **Use with current row as defaults for new entries** - sets the default values for creating new rows in the Study Generator configuration section with the current values on this dialog and on the row containing these settings (the row with the *Settings* button you just clicked).
- **Tag Overrides** - permits setting values in generated images for the Patient’s Name, Study Instance UID, and Accession Number tags. If left blank, reasonable defaults are used.
- **Modality** - sets the modality value for the generated images and a corresponding representative background image. Check the *Black background* box to use a plain, background for generated images.
- **Store Settings** - depending on the Transfer Mode that is selected, either DIMSE Settings or STOW Settings are available that pertain to the current application entry.
- **Apply All** - applies the settings in this dialog to all other Study Generator entries in the scenario.
- **Reset** - resets the settings in this dialog to their defaults.
- **Cancel** - avoid applying these settings to the current Study Generator entry and returns to the main Conductor screen.
- **OK** - accepts these settings and applies them to the current Study Generator instance.

Some settings on the *Study Generator Settings* dialog are disabled. These grayed-out options are not currently configurable, though they will be in a future version of Conductor.

Available configuration options for *HL7 Receivers* are:

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
• **How Many** - enter how many of this application should be run in the current scenario. Maximum is 99.
• **On Server** - the name or IP address of the server on which the application should be run.
• **First Port** - indicates the port on which the first HL7 Receiver running on the specified server will listen on. If more than one HL7 Receiver is to be run, subsequent HL7 Receivers will listen on subsequent ports. If the specific **First Port** is 2675 and four HL7 Receivers are to be run, they will listen on ports 2675, 2676, 2677 and 2678, respectively.
• **Send Ack** - instructs the HL7 Receiver to send an ACK in response to a received message.
• **Description** - an optional comment or explanation added by the user for reference.

Available configuration options for **HL7 Senders** are:

<table>
<thead>
<tr>
<th>HL7 Senders</th>
<th>Run</th>
<th>How Many</th>
<th>On Server</th>
<th>Host Server</th>
<th>First Port</th>
<th>Port Mode</th>
<th>Messages</th>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td>4</td>
<td>localhost</td>
<td>localhost</td>
<td>2575</td>
<td>Same</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>localhost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

• **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
• **How Many** - enter how many of this application should be run in the current scenario. Maximum is 99.
• **On Server** - the name or IP address of the server on which the application should be run.
• **Host Server** - the name or IP address of the server to which HL7 Senders should send the messages they generate.
• **First Host Port** - indicates the port to which the first HL7 Sender running will send the messages it generates. If more than one HL7 Sender is to be run, subsequent HL7 Senders will use the **Port Mode** option to either send messages to the same port or to sequential ports.
• **Port Mode** - the port sequencing mode that determines port numbers used by HL7 Senders. If the **Port Mode** is **Same**, all HL7 Senders will send to the same port indicated by **First Host Port**. If the **Port Mode** is **Sequential**, HL7 Senders will send to sequential port numbers on the **Host Server**, starting with the **First Host Port**.
• **Messages** - total number of messages to generate in the scenario. Maximum is 1,000,000.
• **Settings** - enables configurations using the **HL7 Sender Settings** dialog (below).
• **Description** - an optional comment or explanation added by the user for reference.

The **HL7 Sender Settings** dialog enables several additional options to be configured. While most settings are self-explanatory, details on each are covered in the HL7 Sender section of the PowerTools user manual.
Of note are the following options on the **HL7 Sender Settings** dialog:

- **Use with current row as defaults for new entries** - sets the default values for creating new rows in the HL7 Sender configuration section with the current values on this dialog and on the row containing these settings (the row with the **Settings** button you just clicked).
- **Apply All** - applies the settings in this dialog to all other HL7 Sender entries in the scenario.
- **Reset** - resets the settings in this dialog to their defaults.
- **Cancel** - avoid applying these settings to the current HL7 Sender entry and returns to the main Conductor screen.
- **OK** - accepts these settings and applies them to the current HL7 Sender instance.

Available configuration options for **Command Lines** are:

- **Run** - when checked, an application entry is “active” and will be included when the scenario is run.
- **How Many** - enter how many of this application should be run in the current scenario. Maximum is 99.
- **On Server** - the name or IP address of the server on which the application should be run.
- **Interface** - the interface type of the application, it determines how the application should be run on the server, either as a **Console** application or a **GUI** application.
- **Run With** - determines when the application should be started and stopped, either with other **Clients** (client applications such as Study Generators and HL7 Senders) or with other **Servers** (server applications such as Store Servers and HL7 Receivers).
- **Application** - the complete path to the application executable program to run on the server.
- **Arguments** - command line arguments, if any, to pass to the application when it is run.
- **Description** - an optional comment or explanation added by the user for reference.
**Status Logging Window**

The status logging window at the bottom of the Conductor user interface displays ongoing status messages capturing important events occurring while a scenario is running. Certain key words are highlighted or displayed in bold for emphasis.

![Status Logging Window](image)

**Status Bar**

The status bar at the very bottom of the Conductor interface displays how much time is left in the current running scenario (based on the original Run for duration) and a progress bar that displays overall progress of all requested images being stored from any running Study Generator applications.

![Status Bar](image)
10.4. Conductor Scenarios

10.4.1. What is a scenario?

A Conductor scenario is a collection of servers, applications, and their configurations that can be automatically started and stopped and that generates runtime statistics about their behaviors that can be viewed and analyzed.

For example, a simple Conductor scenario to test Compass throughput could be:

- Conductor - running on its own server, controls starting, stopping and data collection from all other applications in the scenario. The scenario is configured to start a Compass, and then start a single Store Server and a single Study Generator, and run for, say, 10 minutes, gathering stats every 30 seconds.
- Compass - running on its own server, configured to run in Direct mode with a receiver and sender configured to pass images from a Study Generator to a Store Server.
- Store Server - running on its own server, the command line version of Store Server will listen on the port designated by the scenario configured in Conductor.
- Study Generator - running on its own server, Study Generator will send images to Compass which will then send those images immediately to the Store Server in the scenario.

After this scenario runs and completes, the gathered statistics can be viewed in a browser or analyzed from a CSV file. The information available includes:

- Number of images sent by Study Generator
- Number of images received by Store Server
- Calculated throughput from Study Generator to Store Server in GB/hr
- Performance over time of Compass memory use and database table entries
- Performance over time of CPU, Disk and Memory use for each of the servers running each application

10.4.2. Configuring a scenario

The following steps walk you through configuration of a basic scenario, like the one described above. Specific server names and port numbers will be used. Be sure to substitute names and port numbers that are relevant to your specific system.

1. Startup Conductor.
2. Create a Compass entry with server **win2012compass** (or you might need to use its IP address), adding a description if you like
3. Create a Store Server entry to run on **vtsink1** with a First Port of 7001 (or the port you noted or configured)
4. Create a Study Generator entry to run 4 Study Generators each on vtsource1 and vtsource2 with a Host Server of win2012compass, First Host Port of 11112, using a Port Mode of **Same** and some sufficient number of images (e.g., 100 Studies, 1 Series/Study and 10 Images/Series).
5. Test for connectivity using File > View Agent Status to make sure all agents are reachable and running. Start any agent services that are not running.
6. Once all agents are running, click Start. Observe progress and click Stop as needed.
7. Refine the scenario to your liking, and when satisfied with it save it from the File > Save Scenario menu for future use and reference.

You now have a completed scenario that you can load, run and analyze.

10.4.3. Running a scenario from the user interface

The following describes the sequence of events that occur when you load and run a Conductor scenario. Specific events that involve applications in the scenario are only performed on those applications that are “active” (the Run checkbox is checked).

Starting
- Load a pre-existing scenario using File > Load Scenario
- Click Start
- A status message is logged with a timestamp, noting that the scenario is starting. When Use Verbose Status Logging is checked, additional status information will be logged.
- At any point during the startup steps, if a step fails, Conductor cancels the scenario and stops any applications that have already started.
- Conductor verifies that agents in the scenario are alive. For any agents that are not alive, if Try to Start Agents with Scenario is checked, it will attempt to start them on their remote servers, subject to permission restrictions.
- Databases and images directories for all Compasses in the scenario are cleared.
- Databases for all Waypoints in the scenario are cleared if the corresponding Clear Compass with Scenario and Clear Waypoint with Scenario menu options are checked.
- Any configured Compass App Config or Exe Config files are copied to the corresponding servers, and, if any were present, their corresponding Compasses are restarted. Stats reporting timers on all agents for the Compasses in the scenario using the Report every period are started. Conductor verifies that all Compasses are running before continuing. Important: if there is an open Compass Client GUI when the Conductor agent restarts the Compass service, it may be necessary to close and then reopen the Compass Client GUI.
- Any configured Waypoint App Config files are copied to the corresponding servers, and, if any are present, their corresponding Waypoints are restarted. Stats reporting timers on all agents for the Waypoints in the scenario using the Report every period are started. Conductor verifies that all Waypoints are running before continuing. Important: if there is an open Waypoint Client GUI when the Conductor agent restarts the Waypoint service, it may be necessary to close and then reopen the Compass Waypoint GUI.
- HL7 Receivers in the scenario are started and their stats reporting timers are started. Conductor verifies that all HL7 Receivers are running before continuing.
- Store Servers in the scenario, those using both DIMSE and STOW transfer modes, are started and their stats reporting timers are started. Conductor verifies that all Store Servers are running before continuing.
Command Lines running with other Servers in the scenario are started. Conductor verifies that all Command Lines running with Servers are running before continuing.

HL7 Senders in the scenario are started and their stats reporting timers are started. Conductor verifies that all HL7 Senders are running before continuing.

Study Generators in the scenario are started and their stats reporting timers are started. Conductor verifies that all Study Generators are running before continuing.

Command Lines running with other Clients in the scenario are started. Conductor verifies that all Command Lines running with Servers are running before continuing.

Conductor’s Time Left timer is started using the Run for duration and is displayed in the status bar and the saved Stats Logs and HTML Reports are rotated, keeping only the number specified by the Number of Backup Stats Logs & Reports to Keep menu choice.

A status message is logged noting a timestamp for when the scenario started and now long it will run before stopping.

**Running**

Once the scenario is successfully running, the following events occur:

- The agent service on any Compass server in the scenario queries the system and Compass database directly, doing so every period of time indicated by the Report every value, to gather runtime information about CPU, Disk and Memory use, and about numbers of images on the Compass file system and numbers of entries in tables in the Compass database. The agent then sends this information to Conductor via Conductor’s built-in web server.

- The agent service on any Waypoint server in the scenario queries the system and Waypoint database directly, doing so every period of time indicated by the Report every value, to gather runtime information about CPU, Disk and Memory use, and about numbers of entries in tables in the Waypoint database. The agent then sends this information to Conductor via Conductor’s built-in web server.

- The other PowerTools applications (Store Server, Study Generator, HL7 Receiver, and HL7 Sender) each gather their own runtime statistics specific to their function on the same Report every period and send the gathered information to its local agent via the agent’s built-in web server.

- Note that no application-specific statistics are gathered for Command Line applications, although statistics about the system on which they are running are reported by the agent servers, as will all other Conductor-controlled applications.

- For each Report every period that goes by, Conductor requests all gathered information from each agent in the scenario. Each agent sends back its gathered information and clears its own copy of gathered information. Conductor appends this incoming information to its collection of runtime stats it is gathering.

- Conductor backs up all its collected information every period of time indicated by the Backup Stats Log & Report Every value, saving it to the designated data backup directory. This is done primarily as an emergency copy of the gathered data, used in case Conductor or the system it is running on crashes for an unexpected reason during a long-running scenario in order to minimize data loss.

- If the user selects the View Summary menu item, a summarized report of the gathered data is displayed in a text dialog to the user.
• If the user selects the **View Report in Browser** menu item, a more in-depth summary report with numerous graphs of gathered data is displayed in the default web browser on the system running Conductor. By noting the URL and system name, this same browser report can be viewed from another system, as long as the Conductor system is reachable from it, if desired.

• If the user selects the **View Stats Log** menu item, a browser-friendly view of all gathered data is displayed in a scrollable table.

• The links available from the **View Agent Diagnostics** menu item enable the user to directly view the web-based diagnostics interface (below) for each agent service and to verify that the Conductor web server is reachable.

### Agent Diagnostics

On any system running the **Conductor Agent Service**, an Agent Diagnostics interface is available that can be used to troubleshoot unexpected scenario behavior. The following Agent Diagnostics interface can be viewed on a web browser on each system that has a running Conductor Agent Service by pointing the web browser to:

```plaintext
http://localhost:8090/AppManager/
```

<table>
<thead>
<tr>
<th>Diagnostics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run low-level agent commands directly.</td>
</tr>
<tr>
<td>alive - Is the agent alive?</td>
</tr>
<tr>
<td>system - Display agent system information</td>
</tr>
<tr>
<td>processcount - Display process count</td>
</tr>
<tr>
<td>everythingcount - Display aggregate count of all running apps and processes</td>
</tr>
<tr>
<td>Compass [compasstop][compasstart][compaskickstart][compasmonitor][compasstop]</td>
</tr>
<tr>
<td>Waypoint [waypoint-running][waypoint-start][waypoint-stop][waypoint-dbs]</td>
</tr>
<tr>
<td>nuke - Nuke All</td>
</tr>
</tbody>
</table>

9/4/2019 8:04:34 AM

The diagnostics interface has the following options:

• **alive** - displays a status indicating whether the agent is “alive”

• **system** - displays information about the system on which the agent is running, including server name, IP address, OS version, available memory, disk and CPU resources, load on the server, as well as information about any application processes from the current scenario that are running on it.

• **processcount** - displays a number indicating how many application processes from the current scenario are running on it.

• **everythingcount** - displays an aggregate count of the number of applications and application processes from the current scenario are running on it.

• **compass=running** - displays the status of Compass for this agent, including whether or not Compass is installed and whether or not, if present, it is running.

• **compass=start** - attempts to start the Compass service if it is stopped, and displays a status of whether or not the operation was successful and Compass is running.

• **compass=stop** - attempts to stop the Compass service if it is running, and displays a status of whether or not the operation was successful and Compass is stopped.
• `compass=images` - displays information about the Compass image directory on this agent system, its path and how many files and directories are in it.
• `compass=db` - displays a textual table of row counts for all Compass database tables and the connection string used by the Conductor Agent Service to query the Compass database.
• `waypoint=running` - displays the status of Waypoint for this agent, including whether or not Waypoint is installed and whether or not, if present, it is running.
• `waypoint=start` - attempts to start the Waypoint service if it is stopped, and displays a status of whether or not the operation was successful and Waypoint is running.
• `waypoint=stop` - attempts to stop the Waypoint service if it is running, and displays a status of whether or not the operation was successful and Waypoint is stopped.
• `waypoint=db` - displays a textual table of row counts for all Waypoint database tables and the connection string used by the Conductor Agent Service to query the Waypoint database.
• `nuke` - issues the “nuke all” command to the agent to forcefully exit any running applications that the agent is controlling.

**Stopping**

When a failure occurs during the startup sequence, when the Run for time limit expires, or when the Stop button is clicked, Conductor initiates the stopping sequence. The events in the stopping sequence are:

• A status message is logged indicating when and why the scenario is stopping.
• Command Line clients are stopped.
• Study Generators are stopped.
• HL7 Senders are stopped.
• HL7 Receivers are stopped.
• Store Servers, first those running in STOW mode and then those using DIMSE mode, are stopped.
• Command Line servers are stopped.
• Compasses are stopped (just the stats reporting timer is stopped).
• Waypoints are stopped (just the stats reporting timer is stopped).
• The general stats retrieval timer for Conductor is stopped and a final backup of gathered data is written to the Stats Log and HTML Report for the just-stopped scenario.
• Conductor verifies with all agents that all applications really have stopped gracefully and sent back their final gathered stats. If any have not stopped, a Nuke message is sent out that directs any agents with still-running applications to forcefully terminate their lingering applications.
• A final status message is logged with a timestamp indicating that the scenario has stopped.

**Exiting**

When the user decides to exit Conductor, Conductor first checks to see if a scenario is running. If a scenario is running, a dialog queries the user about whether the scenario should be stopped. If the user realizes the scenario should not be stopped, it continues running as before. Otherwise, if
the user requests that the scenario be stopped, Conductor initiates the stopping sequence described above.

When exiting from a stopped scenario, if there are gathered runtime stats that have not yet been saved, a dialog queries the user about whether those stats should be saved. If the user decides to save the gathered stats, a Save File dialog is presented that enables the stats to be written to a CSV file, after which Conductor exits. Otherwise, if the user does not want to save the stats, Conductor simply exits.

10.4.4. Running a scenario from the command line

Conductor can be run from the command line in a very limited way. The steps needed to run Conductor from the command line are:

1. Create config files as needed for any custom Compass configurations you want to use when running the scenario.
2. Create a Conductor config file with the desired scenario configuration.
3. Place all config files in a directory on the same server as Conductor.
4. Determine an output directory, recommended as the same directory as the one where you just placed config files.
5. Use the following command line syntax (all on the same line) to run the scenario:

   ```
   C:\PowerToolsExePath\Conductor.exe
   -config C:\LoadTestConfigsPath\ConductorConfig.xml
   -runtest OutputDirPath
   ```

6. Generated output files will be written in the output directory specified by the `-runtest` argument. Output files consist of the textual and browser-friendly summary reports, textual and RTF version of the Conductor status window, and the final CSV file containing all gathered runtime stats.
7. Results can then be analyzed as desired. For example, subsequent automated runs can be used to compare throughput of Compass under differing conditions or configurations.
8. Two helpful scripts are included in the PowerTools distribution that provide a starting point for customizing your own analyses:
   - `RunConductorAutoTest.bat` - a batch script to automate running a Conductor scenario from the command line and initiating comparison of output results
   - `CompareConductorResults.ps1` - a PowerShell script for comparing output results
10.5. Conductor Troubleshooting

The following are descriptions of some of the potential problems that have been seen when using Conductor and some troubleshooting things to try to resolve them.

<table>
<thead>
<tr>
<th>Description</th>
<th>Things to Try</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conductor Controller won’t start</td>
<td>• Verify that there is a PowerTools license installed</td>
</tr>
<tr>
<td></td>
<td>• Verify that the license includes a conductor feature key</td>
</tr>
<tr>
<td>An application won’t start on one of the agents</td>
<td>• Verify that the agent’s server is running</td>
</tr>
<tr>
<td></td>
<td>• Verify that the agent service on the server is running</td>
</tr>
<tr>
<td></td>
<td>• Verify that all agents in the scenario are reachable: View &gt; Agent Status</td>
</tr>
<tr>
<td></td>
<td>• View the agent’s diagnostic interface for more info:</td>
</tr>
<tr>
<td></td>
<td><a href="http://localhost:8090/AppManager/">http://localhost:8090/AppManager/</a></td>
</tr>
<tr>
<td>General problems with an agent</td>
<td>• Configure the agent to log troubleshooting messages to the Event Viewer by creating a file called .TRACE in the PowerTools application directory, then restart the service: net stop conductor net start conductor</td>
</tr>
<tr>
<td></td>
<td>• Important: when Event Viewer logging is no longer needed, be sure to delete the .TRACE file that you created, then restart the service</td>
</tr>
<tr>
<td>Conductor Agent Service cannot be started</td>
<td>• Check Event Viewer for application error messages for clues to the cause of the problem</td>
</tr>
<tr>
<td></td>
<td>• Turn on .TRACE logging and try restarting the agent service, examining the Event Viewer log for helpful messages. Try online lookup of error messages for possible solutions.</td>
</tr>
<tr>
<td></td>
<td>• If running on a virtual network, verify that a connection to the outside is available if there are network lookup errors in the log. Try re-enabling connectivity and retrying agent startup.</td>
</tr>
<tr>
<td></td>
<td>• Restart the server where the agent is failing to start, then retry agent startup.</td>
</tr>
<tr>
<td>Conductor is not receiving data from an agent</td>
<td>• Verify that you included one or more applications to run on that server in the scenario</td>
</tr>
<tr>
<td></td>
<td>• Verify that one or more applications are running on the server when the scenario is running</td>
</tr>
<tr>
<td></td>
<td>• Verify that the agent is running on the server</td>
</tr>
<tr>
<td></td>
<td>• Verify that the Conductor web server is alive, locally:</td>
</tr>
<tr>
<td></td>
<td><a href="http://localhost:8091/AppManager/">http://localhost:8091/AppManager/</a></td>
</tr>
<tr>
<td></td>
<td>• Verify that the Conductor web server is reachable from the agent server, using the Conductor server name or IP:</td>
</tr>
<tr>
<td></td>
<td><a href="http://conductorserver:8091/AppManager/">http://conductorserver:8091/AppManager/</a></td>
</tr>
<tr>
<td>Description</td>
<td>Things to Try</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PowerTools fails to install</td>
<td>• Stop <strong>Conductor Agent Service</strong> and retry installation: net stop conductor</td>
</tr>
<tr>
<td></td>
<td>• Uninstall current version of PowerTools and retry installation</td>
</tr>
<tr>
<td></td>
<td>• Repair currently installed version of PowerTools, then verify <strong>Conductor Agent Service</strong> is stopped, uninstall current version of PowerTools, and then retry installation of new version</td>
</tr>
<tr>
<td>An application on an agent does not exit</td>
<td>• Run <strong>Tools &gt; Nuke All</strong> command</td>
</tr>
<tr>
<td></td>
<td>• On the agent server, manually terminate the application</td>
</tr>
<tr>
<td>Agent Diagnostics</td>
<td>• On the agent server, open the interface in a web browser:</td>
</tr>
<tr>
<td></td>
<td><a href="http://localhost:8091/AppManager/">http://localhost:8091/AppManager/</a></td>
</tr>
<tr>
<td>Conductor Web Server Reachability Test</td>
<td>• On the Conductor server, open the sanity check web page in a web browser:</td>
</tr>
<tr>
<td></td>
<td><a href="http://localhost:8090/AppManager/">http://localhost:8090/AppManager/</a></td>
</tr>
<tr>
<td><strong>Using Compass with Conductor</strong></td>
<td>• Restart the Compass Client</td>
</tr>
<tr>
<td>Compass Client GUI unexpectedly closes after a Conductor scenario starts</td>
<td>• If the Compass services was restarted by the Conductor agent, it may need to be restarted to properly connect to the Compass database</td>
</tr>
<tr>
<td>(same for Waypoint)</td>
<td>• Configure Compass to automatically start incoming and outgoing connections for DICOM and HL7 so they will automatically start after a Compass service restart. This is done on the <strong>System</strong> tab by checking <strong>Enabled at Startup</strong> for each.</td>
</tr>
<tr>
<td>Compass doesn’t start incoming and outgoing connections after a Conductor</td>
<td></td>
</tr>
<tr>
<td>scenario starts</td>
<td></td>
</tr>
<tr>
<td>A Compass Source doesn’t accept incoming traffic, but seems like it should</td>
<td>• Verify the port you are sending to is the same port set in the corresponding <strong>Listener</strong> in Compass.</td>
</tr>
<tr>
<td></td>
<td>• Consider checking <strong>Accept All</strong> for AE Titles on Sources if AE Title matching is not needed.</td>
</tr>
</tbody>
</table>