SWITCHBOARD USER MANUAL

Applies to Switchboard version 3.5.4c

LAUREL BRIDGE

Providing DICOM Connectivity for the Medical Community

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

Document Version: 3.5.4c
Document Number: LBDC-000060-0354
Last Saved: 10/11/2018 2:51:00 PM

What's recently been added to the manual?

- Updated application screenshots.
- Clarified notes on using filters.
- Updated firewall information.
- Described how to use the "Element Composer" filter.
- Described how to check if a tag exists for "Elements to Match" filter, or how to filter only if a tag does not exist.
- Updated Linux installation steps.
- Updated information on the Mapping Rule Editor and Filter Set Editor and the other web applications.
- Updated hostname resolution information.
- Described how to use the Planar Configuration Convert filter.
- Updated License Activation information.
- Described changes needed for Jpeg2000 compression fractional bits.
- Updated instructions for upgrading from an older version.
- Described how to define a custom filter.
- Added information on Session Settings.
- Updated instructions on selecting a filter.
- Described how to change the web port.
- Updated the software installation requirements information.
- Added appendix on Conditional Filters.
- Added appendix on specialized filtering options.
- Updated information on duplicating a configuration.
- Added information on manual configuration to recompress image data.
- Added information on how to fix invalid Icon Image Sequences.
- Described startup scripts for SUSE Linux 13.2.
- Described upgrade process for Linux.

Table of Contents

Table of Figures

1 Using the DICOM Switchboard

As you integrate various medical devices that use the DICOM protocol to communicate with each other, situations may arise in which some of the data being sent from the source to the destination is not entirely correct in some way – it may need to be in a different format, is expected in a different DICOM tag, or some other problem may need to be corrected. The DICOM Switchboard can help in many of these situations by filtering the data as it is sent from the source device to the destination device. In addition, the DICOM Switchboard provides a state-of-the-art DICOM protocol monitoring capability that can assist you in diagnosing DICOM connectivity issues.

To do these tasks, the DICOM Switchboard is interposed between the source and the destination devices, and any communication between the two devices goes through the DICOM Switchboard.

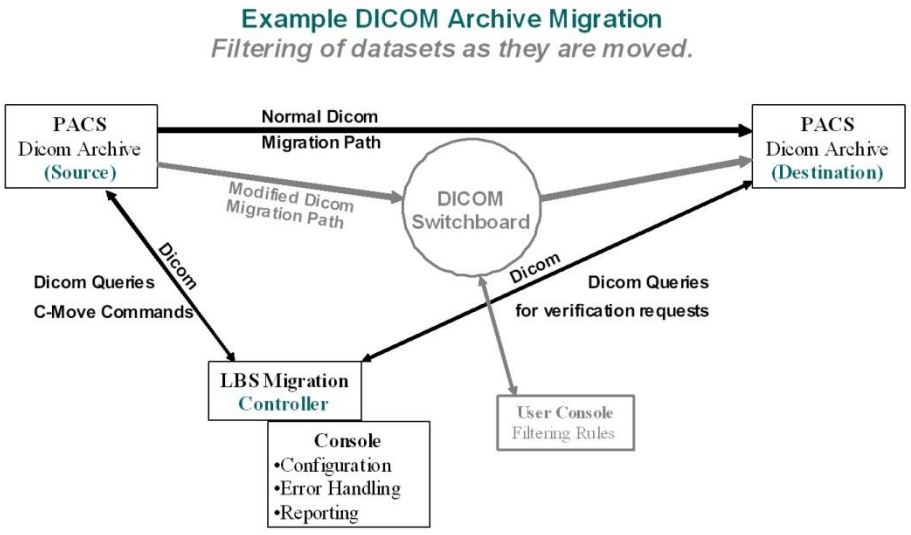


Figure 1 – DICOM Communication with Filtering Diagram

The source device (e.g., the modality) is configured to send its DICOM data to the DICOM Switchboard, which is configured to forward the filtered data on to the destination device (e.g., the archive). Note: it is possible that the DICOM Switchboard could be installed on the same box as the source device, however this is not required; the DICOM Switchboard could be on another box entirely or even on the destination host itself.

The DICOM *communication* between the source and the destination goes through the DICOM Switchboard. The source is configured to send to the DICOM Switchboard. The DICOM Switchboard's Mapping Rules – configured by the user – tell it where to send any data it receives; these rules would normally be set to forward the data to the destination. You can create filters that the DICOM Switchboard will use to modify the data received, and then the DICOM Switchboard will send or forward the data on to the destination. The DICOM Switchboard has the capability to handle many different mapping and filtering scenarios simultaneously.

Note that the DICOM Switchboard is **not** a "network packet sniffer", passively listening for communication on the network but not actively participating in it. The Switchboard is an ***active participant*** in any DICOM communication that goes through it. You configure your SCU to

communicate with the Switchboard, and the Switchboard is configured to communicate with your SCP. To put it another way, the actual SCU thinks that the Switchboard is the target SCP, while the actual SCP thinks that the Switchboard is the desired SCU.

[**SCU**] → [**DICOM Switchboard**] → [**SCP**]

The SCU communicates to the SCP via the DICOM Switchboard; the Switchboard acts as the SCP to the SCU and as the SCU to the SCP. The mapping rules tell the DICOM Switchboard where it should send the messages it receives – some can go to one place, and some can go to another. Only by being an active participant can the Switchboard modify and filter the DICOM data.

Note also that the DICOM Switchboard is **not** a store-and-forward device – the DICOM data is never physically stored by the DICOM Switchboard, it is just held in memory long enough to be filtered, and then it is sent on to the destination.

2 Overview: Basic Steps to Implement DICOM Monitoring and Filtering

Follow these steps to set up the DICOM Switchboard to monitor DICOM transactions and to filter DICOM datasets:

1. Install the DICOM Switchboard on the desired system. This could be the source host, destination host, or some other host, but it must be network accessible to the source and destination devices.
2. Configure the Switchboard's various communication settings using the "*Configure Switchboard*" option from the Switchboard Operations Window. You may need to modify your firewall's configuration for open ports and allowable applications – this can include communication with the Switchboard's web server as well as with the Switchboard itself.

   Chapter **7 Configuring the DICOM Switchboard** has detailed information.
3. Set up the filters required for modifying data, if any.

   Chapter **8, Filters and Filter Sets**, has a guide to the filters available and how to configure them to do what you want
4. Set up the mapping rules for the DICOM Switchboard. If necessary, associate filters with each mapping rule.

   Chapter **9 Switchboard Mapping Rules** describes how to create the mapping rules.
5. Configure the source to send to the DICOM Switchboard. This may require setting permissions in the source host to allow communication with the Switchboard's host and AE-Title.
6. Configure the destination to accept from the DICOM Switchboard, if required.
7. Test the setup to confirm that data is following the desired path and is being filtered as intended.

3 Installing the DICOM Switchboard

Note: The DICOM Switchboard is used primarily on Windows platforms, but it is available for Linux users as well. Most of the steps to operate the DICOM Switchboard will be the same on all platforms, but some steps may vary for Linux; these exceptions will be noted throughout this document.

3.1 Prerequisites:

Before completing your install of the DICOM Switchboard, make sure you have the following applications installed and configured on your system. Most of these components are included in the DICOM Switchboard installation zip file or can be freely downloaded from the Web. As noted below, the DICOM Switchboard Installer for Windows will check your system for these prerequisites and offer you the option of installing any that are not already installed.

Additional information for Linux installation is found in **Section 3.4** below.

- **Software**

  The DICOM Switchboard is normally provided for installation on 64-bit Windows (including Windows 7, Windows Server 2016, and newer), but can be supplied for a variety of popular platforms, including various 64-bit Linuxes.

- **Hardware**

  Hardware that supports the underlying operating system should be sufficient for running the DICOM Switchboard. Increased memory and multiple network cards (NICs) can improve application performance.

  A recommended system configuration can be seen on the Switchboard product page of the Laurel Bridge website: http://www.laurelbridge.com/switchboard.html.

- **JRE – Java Runtime Environment** (or the Java SE SDK)

  http://www.java.com

  Recommended version: ***8 Update 72 or later***

  Java will automatically be installed by the Windows installer; Linux users should install a **64**-bit (x64) Java manually as one of their Linux installation options. Note that you will need *at least* Java 7 if you have a Product Serial Number for license activation (see Section **13 License and Activation, step 3)**.

- **Apache web server**

  http://www.apache.org/dist/httpd/

  Recommended version: ***2.2.16***

See **Appendix F: Section 3, Using Apache 1.3 on Linux** for help in compiling Apache 1.3 for Linux if you want to use that version.

- **Perl**

  Recommended version: 5.8.6

  Perl will automatically be installed by the Windows installer; Linux users should install it manually as one of their Linux installation options, or by downloading and building the appropriate version and making sure that Perl is in the PATH.

- **GCC 4.9.3 Libraries** (for Linux installations only)

  The Switchboard is built with GCC 4.9.3 for 64-bit Linux, verify that the correct GCC libraries for your target operating system are present or install them if they are not.

- **Web browser**

  You must also have a web browser installed in order to control the DICOM Switchboard via the Web and to use its web applications to monitor the data. It is recommended that you use a modern browser, such as Internet Explorer (v.8 or later) or Firefox (v.10 or later); older browsers may have issues with JavaScript in the web applications.

  Almost any standard web browser can be used to control the DICOM Switchboard and its operations. This is done via web applications, CGI scripts, and standard web pages with JavaScript. Note that JavaScript *must* be enabled in order to use the pages properly. Due to differences in the implementation of JavaScript in browsers, some older browsers may not be able to display the pages correctly (for example, older versions of Linux's Konqueror browser).

  The DICOM Switchboard has been tested successfully on the following browsers:

  - Internet Explorer 8 and later on Windows
  - Firefox 12 and later on Windows
  - iPad

- **PDF Reader**

  Many supplementary documents in the DICOM Switchboard are PDF (Portable Document Format) files. You should download and install your preferred PDF reader (e.g., Adobe Acrobat Reader DC) in order to view the documents.

For Windows, the DICOM Switchboard Installer – **Setup.exe** – will check your system for prerequisites that are not installed and offer you the option of installing them.

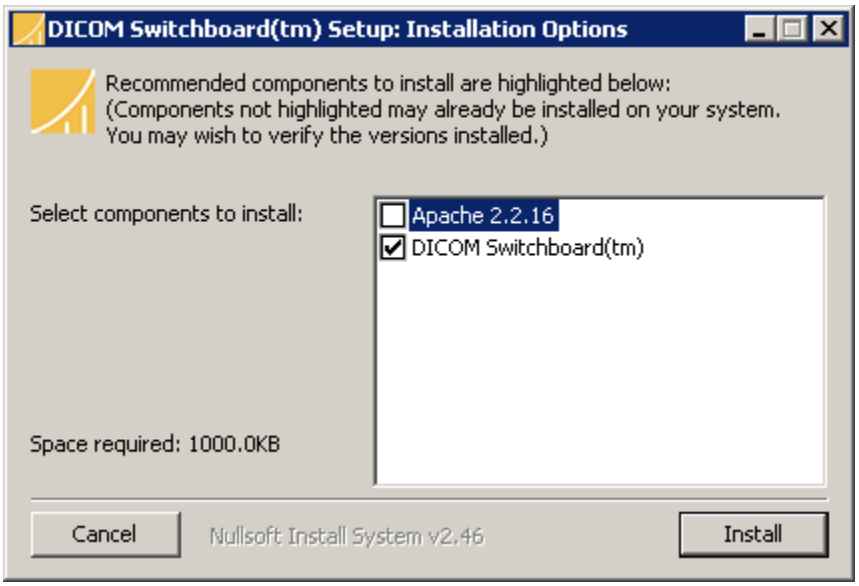


Figure 2 – Selecting the Prerequisites to Install

You may wish to note where these applications are installed, as you will need to know this information in Step 4) below of the Windows installation process described in Section 3.3.

IMPORTANT NOTE ON SOFTWARE UPDATES:

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

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

We also recommend that automatic updates be disabled on systems; while we encourage updates, especially security updates, before deploying new updates we do recommend testing and manual application of such updates.

A system administrator should manage and be present for the application of any upgrades and for any system re-boot – for whatever reason. Be wary of unintended consequences like privileges, permissions, or firewalls that change as a side-effect of patches.

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

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

3.2 Notes on Installing Apache

If you install Apache with the DICOM Switchboard, you can choose to install it as either a service (for every user) or to start manually (for the current user) – see **Figure 3** below. In manual mode Apache is automatically started for you whenever you start the DICOM Switchboard. If it is installed as a service, the Switchboard can still start Apache correctly for its own use – note that this is not the same as configuring Apache to run as a service for the Switchboard. Apache – when installed as a service – can be configured for the Switchboard's use to be started automatically **or** manually.

It is preferred that Apache be installed to start manually, but this is not required. Apache can be installed to start manually, and the Switchboard will be configured separately, including the option to start Apache automatically for the Switchboard **only** (see **Figure 5** below). See section **3.5.2** below and Chapter **15, DICOM Switchboard Web Server** and **Appendix F: Additional Information on Apache**, Section **1: Configuring Apache to run as a Service** for more information.

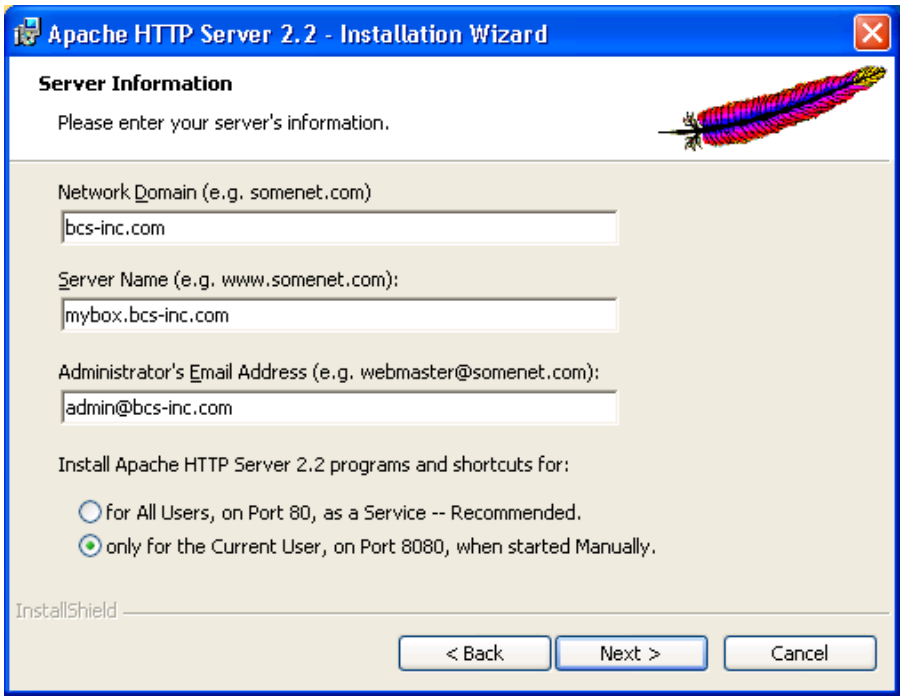


Figure 3 – Installing Apache

3.3 Installing the DICOM Switchboard on Windows

Please note that you should be logged in as the Administrator (or a user with Administrator privileges) when installing the DICOM Switchboard – otherwise the system may not be able to access all the system resources it needs when installing and operating. (See the note at the end of this section for information on how user permissions can affect the ability to access the Switchboard.)

If you have a previous version of the Switchboard installed, you should uninstall it before installing the new version of the Switchboard. (See Chapter **3.7** for help if you are upgrading to a newer version of the Switchboard.)

Follow these instructions to setup your system to use the DICOM Switchboard.

1) Install the DICOM Switchboard and its support software by running **Setup.exe**. (To only install the DICOM Switchboard software and not the support software, run the separate Switchboard installation program, **Install_Switchboard.exe**.)

   The installer will attempt to determine if necessary software, such as Apache, is installed on the computer. If their presence cannot be confirmed, you will be asked if you wish to continue with the installation. You may choose to continue installing the DICOM Switchboard and then install the other necessary software later, or you may cancel the Switchboard installation to install the necessary support software first.

2) You will be prompted to choose a directory for the install, for instance, "`C:\LB Switchboard`". Enter the desired directory in the dialog box, or select whatever directory you prefer.

   Note: If you are installing on Windows 7 or Windows Server 2008 or higher, you may need to disable the UAC (User Access Controls) in order for the DICOM Switchboard to be installed correctly. It is **not** recommended that you install in "`C:\Program Files`"; choose an alternate installation directory, such as "`C:\LB Switchboard`" instead.

3) The DICOM Switchboard and its associated software require a valid run-time license to operate correctly. The license installation key is typically downloaded and is stored in a file named with the "**.key**" extension, e.g.,

   `DSB-3.5.2c-DM-company-site-host-YYYYMMDD-xx.xx.xx.xx.xx.xx.key`

   Press the "**…**" button to select the license key file to use and then press the "*Next*" button to continue with the installation.

   Alternatively, if you have a Product Serial Number for an Activation-style license, click the checkbox to activate it later in the installation, and then press the "*Next*" button to continue with the installation.

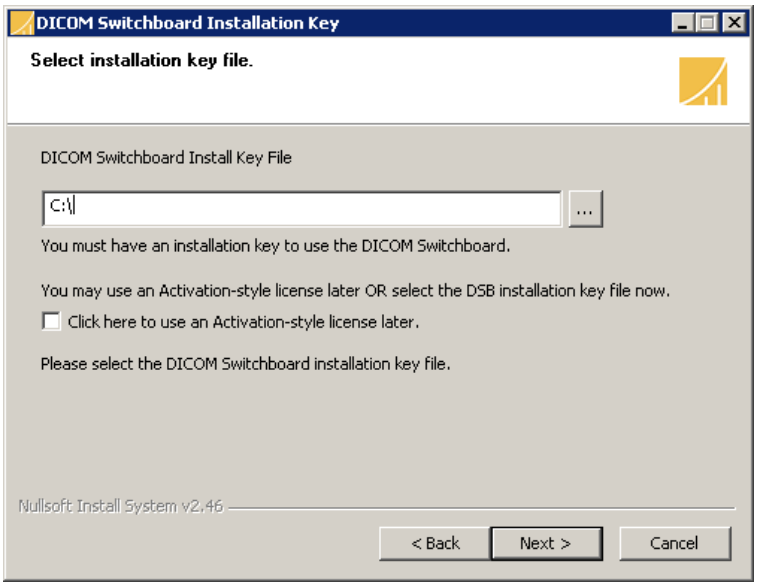


Figure 4 – Selecting the license installation key

4) Please enter the path of the Apache component (shown in **Figure 5** below). You may browse your computer to locate this component by pressing the "**…**" button next to the box. Select a port for the Switchboard's Apache Web server to use (be sure another application is not already using this port). You may choose to install the DICOM Switchboard as a Windows service, if you wish for the Switchboard to be started when you start Windows. If you are

installing the DICOM Switchboard as a service, you can also optionally choose to configure the Apache web server as a service.

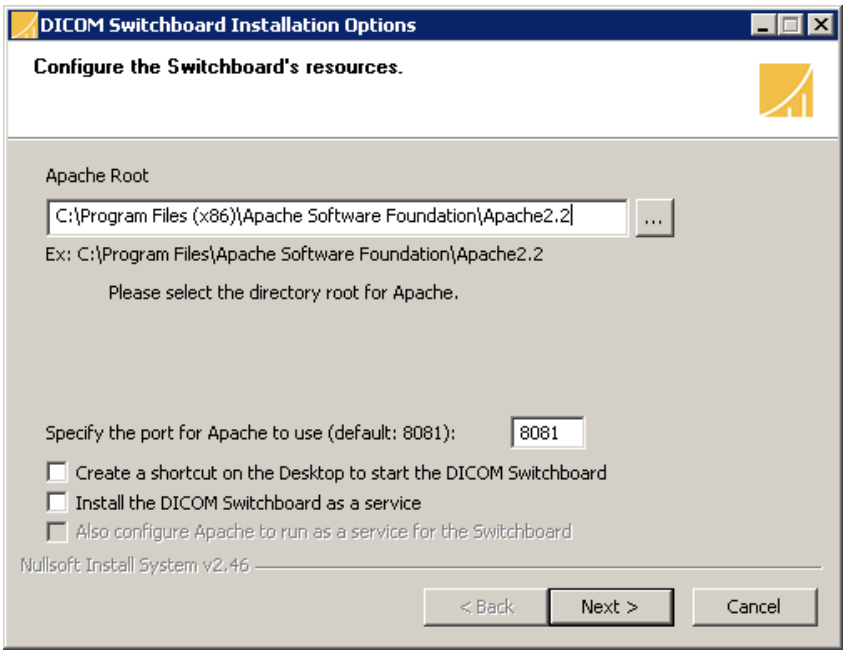


Figure 5 – Configuring the Switchboard's Resource Locations

Note: The Apache port that you select here is used to control the DICOM Switchboard via its web interface. The Switchboard has a *separate* port that is used for communication with your PACS and your modalities – this is configured after the DICOM Switchboard is installed (see **Configuring the DICOM Switchboard** in Chapter 7 below).

5) After verifying the configuration data, the installer will copy the DICOM Switchboard files to your system and then configure the Switchboard and set up its system environment. These configuration steps may take a few moments, so please be patient as you wait for the setup commands to finish…

6) If your license requires activation or you were given only a Product Serial Number, you must activate the license before you can use the Switchboard. If this is so, you will be given the option of activating the license during the installation – you can also choose to activate it later, via the Windows Start menu (note that you may need assistance from Laurel Bridge Software if you are activating later). If you choose to activate it now, you will see the License Activation Utility – fill out ***all*** the fields (only the MAC Address is optional) and press the "Activate" button. (Note that the License Activation Utility can take several seconds to start the first time that it is run, so please be patient.) Once the license is successfully activated – or if you change your mind and want to activate the license later – exit the utility by pressing the Exit button. The installation will continue. (See Section **13 License and Activation** for more information about activating your license, especially if choose not to activate it now.)

7) The installation will prompt you to reboot the computer. The computer must be rebooted in order for the operating system environment changes to take effect.

Figure 6 below shows the structure of the directories that are created when you install the DICOM Switchboard. This can help you if you need to edit a configuration file manually for advanced applications of the DICOM Switchboard.

Note on User Permissions: When you install the Switchboard, you should do it as Administrator or a user with administrative privileges so that it can access the various system resources it needs. After that, when the Switchboard is running, anyone can access its web page, subject to whatever restrictions you may choose to place on the Switchboard's web pages (see **Appendix F: Authenticating Access to Switchboard Web Pages** for information on doing this).

If you logout of the computer and then a different user logs in, he may or may not be able to access the Switchboard's Start menu options and to start or stop the Switchboard's web page. If the user does not have administrator-level permissions, she will probably not be able to start Apache or the Switchboard. If the user does have administrative privileges, he probably will be able to access the Switchboard's Start menu but still may not be able to start Apache or the Switchboard.

This can depend on the OS version and the permissions assigned to each user. For example, in some versions of Windows, the permissions assigned to users and groups are so restrictive that only the user who installed the Switchboard can start it, unless Switchboard is installed as a service.

You can also change the permissions assigned to a user for accessing a directory or even add a user to let him/her access it. (See your OS's instructions for help on how to do this.) In such a case, you should give the user permissions to read, write, and modify files in the Switchboard directory and in all its subdirectories; this will allow the Switchboard application to create the necessary files (such as log files) required for it to operate.

Note on Anti-Virus: The above concerns may also apply to various anti-virus products. You should make sure that your anti-virus is not blocking Switchboard's servers from starting or operating correctly; also make sure that the anti-virus is not preventing access to Switchboard's DICOM or web ports.

3.4 Installing the DICOM Switchboard on Linux

The installation steps are described fully in the file **`README.linux.Switchboard_setup.txt`**; they are summarized here.

1) Go to the directory where you wish to install the DICOM Switchboard, e.g., `/opt`. Note that you may need to be `root` to install in `/opt`, but you should not be `root` when running the setup procedure.

2) Extract the DICOM Switchboard's files from the "tarball":

```
tar xzvf Switchboard-3.5.2c-linux.tgz
```

3) Change (cd) into the Switchboard's installation directory, e.g., `/opt/DSB-3.5.2c`, and **edit the `app_platforms.cfg` file**, setting the values as appropriate.

   Note: You should select the section of the config file to edit based upon the version of the Switchboard in use and the Linux OS it is using. For instance, select "Linux_13_x64_gcc_493" if you are using 64-bit Linux with GCC 4.9.3.

   Settings to change:
   DCF_ROOT should be set to the directory where you installed the Switchboard;
   JAVA_ROOT and

JAVA_BIN should be wherever you have already installed the Java Runtime Environment (JRE), and similarly for Apache.
Be sure to uncomment and set the value for the APACHE_MODULE_DIR.

> Note: **APACHE_ROOT** is the directory where Apache is installed; it should have subdirectories with the Apache modules, configuration files, etc.; this could be, for example, "`/opt/apache-2.2.16`". **APACHE_BIN** is the directory where the Apache executable – usually called "*httpd*" – can be found; this directory may simply be found under the APACHE_ROOT directory, but this is not necessarily true on all versions of Linux. **APACHE_MODULE_DIR** is the directory containing any Apache modules that are used; this is often "`APACHE_ROOT/lib`" or "`APACHE_ROOT/libexec`".

Set the value for LD_LIBRARY_PATH to be the directory where the standard C++ link libraries are. (The Switchboard is built with GCC 4.9.3 on 64-bit Linux, so the LD_LIBRARY_PATH is the directory where the correct version GCC libraries are located.)

Select a port for the Switchboard's Apache Web server to use (be sure another application is not already using this port).

4) Install your key file by copying it as follows (the filename will become systeminfo):

```
cp <keyfile> cfg/systeminfo
cp <keyfile> devel/cfgsrc/systeminfo
```

Note that you must already have gotten your license key file from Laurel Bridge Software – for example, e-mailed from LBS or downloaded from the LBS license site.

If, instead of a license file, you received a Product Serial Number for license activation, complete that in Step **8**) below.

5) From the installation directory (instdir), run this command to set up the Switchboard's configuration files and environment (substituting the appropriate value for "instdir"):

```
perl bin/dcfsetup.pl -zip -web -noprompt
-platform_file_name app_platforms.cfg -envfile dcf.env
-dcf_root=<instdir>
-platform_name Linux_13_x64_gcc_493
-jar_cp=<instdir>/classes/LaurelBridge.jar:<instdir>/classes/DFT.jar
 -f
```

Note: Specify as the platform_name the name of the section that you edited in the app_platforms.cfg file in Step **3)** above.

6) "Dot" the dcf.env file to set the environment. (**`. ./dcf.env`**)

7) Update the Switchboard's configuration files for your environment:

```
perl bin/update_cds.pl -f
```

8) If your license needs activation, you need to activate it now before the Switchboard will run.

To activate a license, run one of the commands below (note that this license activation utility must be run with the Switchboard's environment variables set, as shown in Step **6**) **above**):

- `bash Run_ActivateDcfLicense.bash`

or

- `java com.lbs.SwitchboardActivation.SwitchboardActivation`

Enter all the necessary fields – your Product Serial Number may have been sent to you via e-mail, and the MAC address is optional – and press the Activate button. If an error occurs, correct the error and try again.

If you are using a different version of Apache than 2.2.16 or a different web server, consult **Appendix F**:, the section **Using an Alternate Web Server** for help on how to configure your web server for Switchboard.

Note: You may need to add "write" permissions to the "cfg" and "tmp" directories, among others, in the Switchboard's installation directory since Apache usually runs as "nobody", with limited permissions. This depends on who the Switchboard is set to run as and the permissions of the Apache web server. However, if the Switchboard and Apache are configured to run as members of the same group, you may not need to grant "universal" write permissions to those directories – just grant write permissions to those users and that group, e.g., "-rwxrwxr--".

Note that when using the DICOM Switchboard on Linux, most commands will need to be run from a command terminal that has the Switchboard's environment set by "dotting" the dcf.env file. This is equivalent to the "Switchboard Command Prompt" for Windows and will be referred to as a "**Switchboard Linux Command Terminal**".

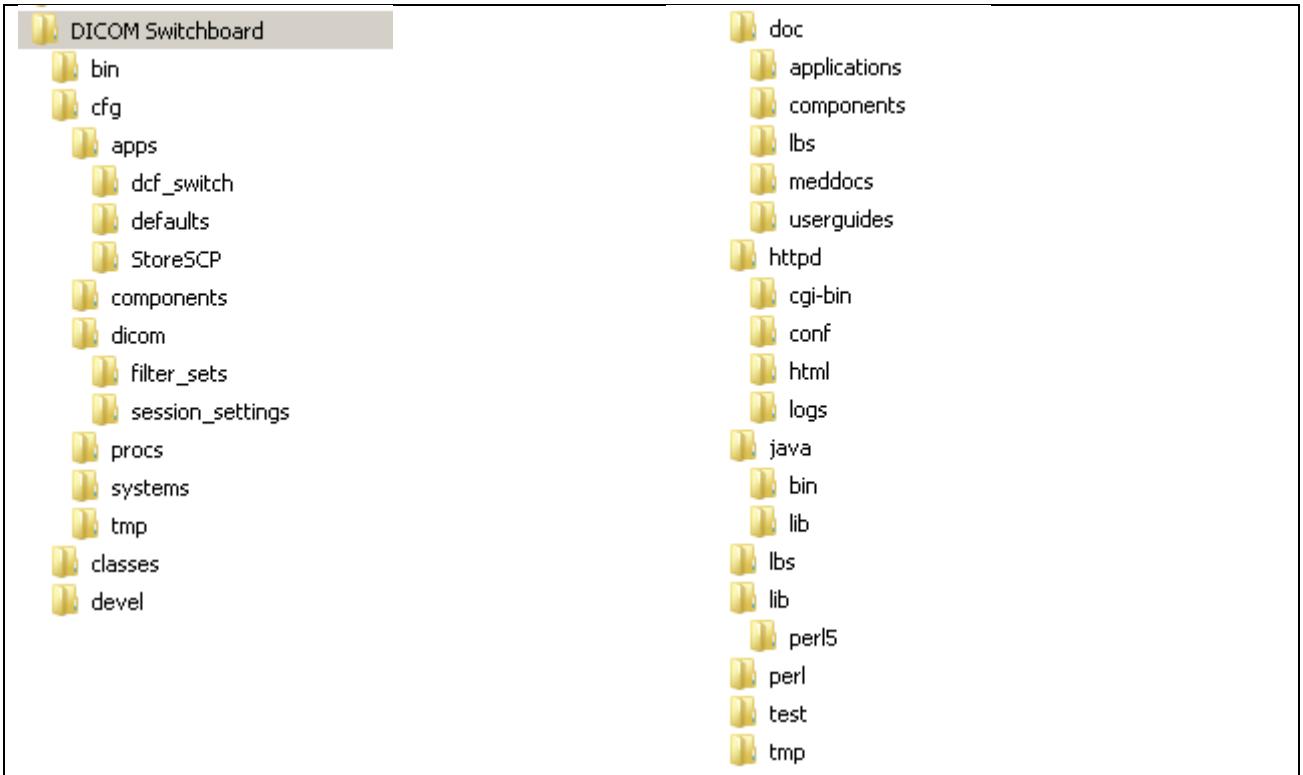


Figure 6 – Directories installed with the DICOM Switchboard

3.5 Installing the DICOM Switchboard as a service on Windows

You may choose to install the DICOM Switchboard as a Windows service.

The service option may not be desirable for an intermittent user – someone who is using the DICOM Switchboard occasionally and does not want the Switchboard to be running continually. On the other hand, if the Switchboard is to be installed as a dedicated application, then the "service" option may be advantageous.

If you choose the "Install the DICOM Switchboard as a service" option, then it will start automatically when its host computer is started. If you do not want it to be started automatically, you can change its start mode from automatic to manual via the Windows Control Panel. To do this select:

`Start → Control Panel → Administrative Tools → Services → DSB.X.Y.Z.dcfsysmgr`

and set the "Startup Type" option under the Properties menu to "Manual".

You may also stop, start, or restart the DICOM Switchboard service from the Windows Services panel. This can be seen in **Figure 7** below.

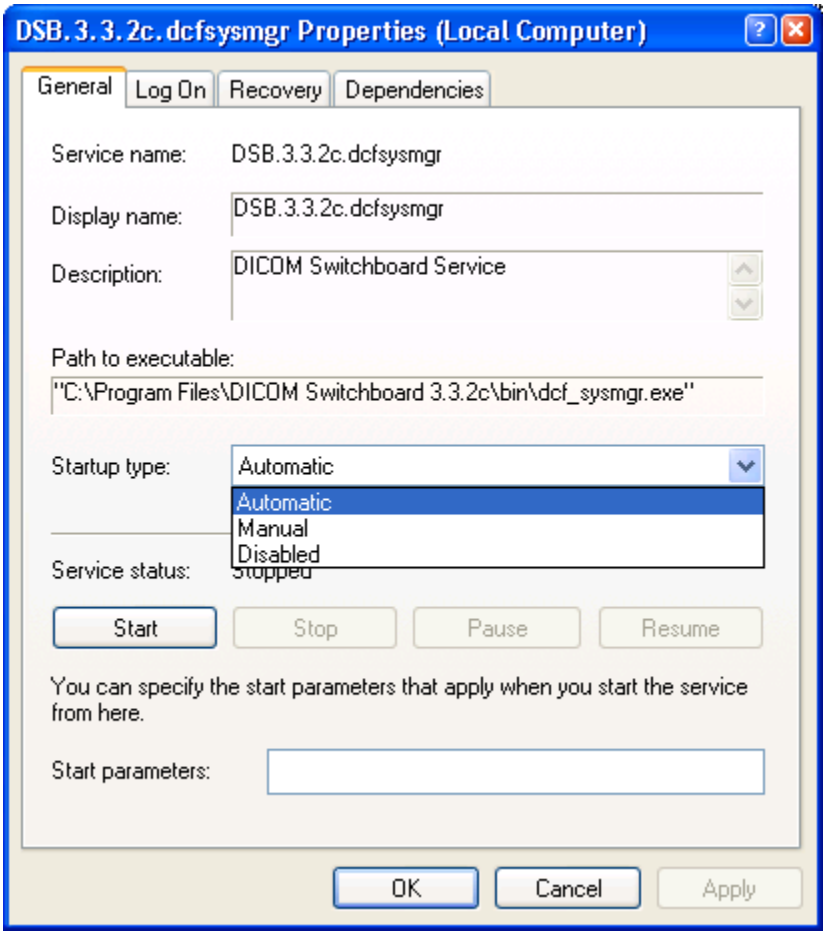


Figure 7 – Changing the Switchboard service's startup mode

Notes on Configuring the Switchboard service to Restart:

When you install the Switchboard as a service, it will automatically be configured to restart itself if an error occurs that causes the Switchboard or one of its supporting Java servers to fail. However, if

it fails more than 3 times, this may mean a serious error is occurring repeatedly, and the problem should be investigated more thoroughly before you restore the Switchboard to normal operations. (Note: The number of restarts is configurable by editing the configuration file for dcf_sysmgr in `%DCF_ROOT%/cfg/apps/defaults/dcf_sysmgr` and setting the value for `max_auto_restarts` – use the value **-1** if you want the Switchboard to restart an indefinite number of times. You will need to restart the dcf_sysmgr service if you change this value.)

If an error occurs in the DCF System Manager (dcf_sysmgr) itself (not in the Switchboard or its Java servers), you may wish to configure the service to restart. This can *not* be done by the Switchboard's installer but must be done *manually* on the Properties panel of the Windows Services page (as shown in **Figure 7** above). Open the panel and select the Recovery tab. Set the actions to be taken on each failure of the Switchboard service and then click "Apply" at the bottom. (This is shown in **Figure 8** below.)

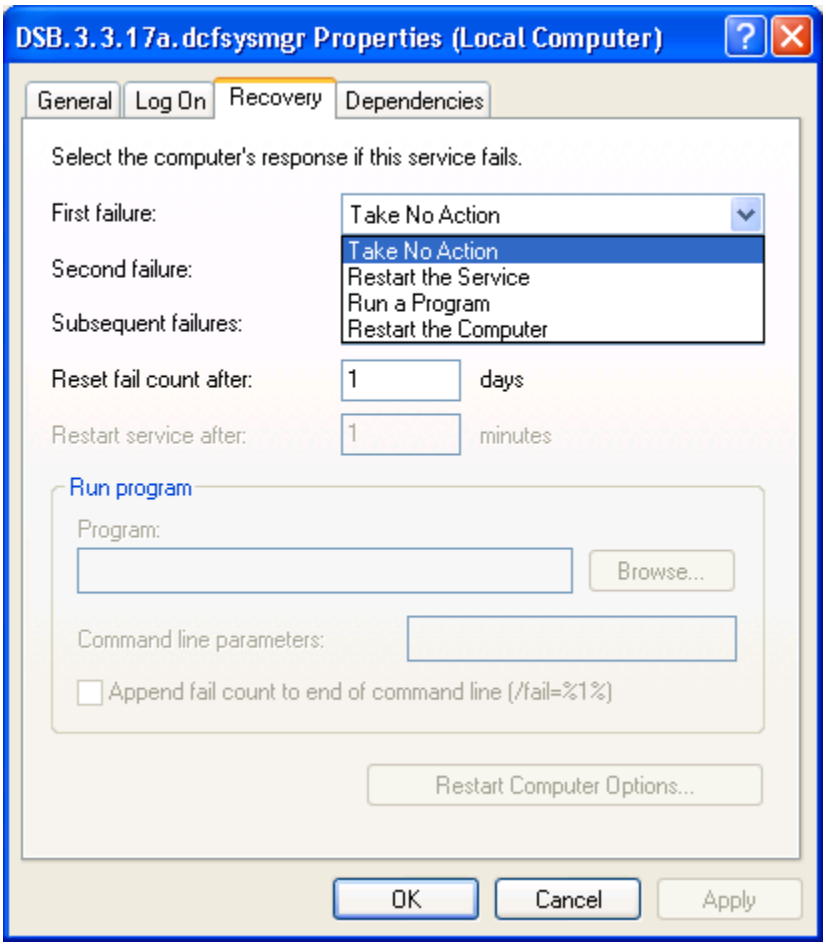


Figure 8 – Setting failure actions in the System Manager service

Miscellaneous Notes on the Switchboard Service:

When you install the DICOM Switchboard as a service, Apache does not get installed as a service and so the web interface to the DICOM Switchboard will not automatically be available. Under this scenario, if you start the DICOM Switchboard via the Start menu (see chapter 4 below), Apache will be started and you will then be able to access the Switchboard's web interface. After you have configured the DICOM Switchboard, you can disable the Switchboard web interface if you wish (see Chapter **15, DICOM Switchboard Web Server**, below for more information about this).

If the DICOM Switchboard is left as a manual start service, then you will have to manually start the service every time you want to access the Switchboard's web interface.

Note: If you chose the "Install the DICOM Switchboard as a service" option, then when you uninstall the DICOM Switchboard, you should stop the DICOM Switchboard service and change it to a manually started service, via the Windows Control Panel and the Administrative Tools, before the DICOM Switchboard can be completely removed from your system.

3.5.1 Configuring the Switchboard service on Linux

To run the Switchboard as a service (or "daemon") on Linux (e.g., SUSE 12), you will need to create a startup script (usually in `/etc/rc.d` or a similar directory). The script should set the necessary DCF environment variables (these can be found in the `dcf.env` file in the Switchboard's installation directory) and export them. The following commands should be included in your startup / shutdown scripts:

To start the Switchboard:

```
$DCF_ROOT/bin/dcfrestart.pl -c $DCF_CFG/systems/dsb_switch_unix.cfg
```

To stop the Switchboard:

```
$DCF_ROOT/bin/dcfstop.pl
```

For SUSE Linux 13.2:

Login as root.

Create the file `/etc/systemd/system/switchboard.service` with the following contents, changing `<INSTDIR>` to be your complete path to Switchboard's installation directory, e.g., `/opt/DSB-3.5.3c`.; also, set `<USERNAME>` to be the user that you want Switchboard to run as (this should probably be the same user that Apache uses). An example of this file is included with the distribution.

```
[Unit]
Description=SwitchBoard Service
After=systemd-user-sessions.service

[Service]
Type=forking
#PIDFile=
User=<USERNAME>
GuessMainPID=true
ExecStart=<INSTDIR>/switchboard_ctl.sh start
ExecReload=<INSTDIR>/switchboard_ctl.sh restart
ExecStop=<INSTDIR>/switchboard_ctl.sh stop

[Install]
WantedBy=multi-user.target
```

Under your Switchboard installation directory, create the file `switchboard_ctl.sh` if it doesn't already exist, with the following contents; make sure that the file has execute permissions.


```
#!/bin/bash
CMDDIR="$( cd "$( dirname "${BASH_SOURCE[0]}" )" >/dev/null && pwd )"
CMD=$1
. "$CMDDIR/dcf.env"

case $CMD in
start)
    exec "$DCF_BIN/dcfstart.pl" -c "$DCF_CFG/systems/dsb_switch_unix.cfg"
    ;;
stop)
    exec "$DCF_BIN/dcfstop.pl"
    ;;
restart)
    exec "$DCF_BIN/dcfstart.pl" -c "$DCF_CFG/systems/dsb_switch_unix.cfg"
    ;;
*)
    echo "option($CMD) is unknown"
    exit 1
    ;;
esac
```

Once these files are created, you can use these commands to start or stop Switchboard:

- Command to start switchboard as test:
  `systemctl start switchboard`
- Command to stop switchboard as test:
  `systemctl stop switchboard`
- Command to set switchboard to start at boot time:
  `systemctl enable switchboard`
- Command to view logs from systemctl:
  `journalctl -u switchboard -f`

Note that these commands should be run as `root`.

You should also modify the configuration values in the `dcf_sysmgr` configuration file. (The file is found in `$DCF_ROOT/cfg/apps/defaults/dcf_sysmgr`.) All the values to be modified are in the "`cpp_ipc_app/dcf_sysmgr`" section. Set the following values:

- `auto_start_system_cfg = /systems/dsb_switch_unix.cfg`
- `exit_after_system_stopped = NO`
- `exit_after_system_error = NO`
- `restart_after_system_error = YES`

The first value tells the Switchboard's System Manager the name of the configuration of Switchboard processes to start. The next two settings tell it that it should not exit if a failure occurs in one of those processes, while the last value tells it to restart after such a failure. (Some of these values are applicable only if `auto_start_system_cfg` is set.)

3.5.2 Running Apache as a service

If you installed the DICOM Switchboard as a service, you can choose to configure Apache to run as a service, too. This means that Apache will be started automatically when the computer starts and that the Switchboard's web interface will be available. This option may not be desirable for the intermittent user – someone who is using the DICOM Switchboard occasionally and does not want the

Switchboard to be running continually. And even if you are running the Switchboard as a dedicated application, you may not want the web interface to be available constantly – you might want to use the web interface to configure the Switchboard but then turn Apache off to prevent others from accessing the configuration options.

But if you do want to have the Apache web interface available all the time, you can select this checkbox, and Apache will be configured as an auto-starting service. If you want the service to be started manually only, you should change it via similar steps as those described above for the DICOM Switchboard service. Note that this option is only available if the DICOM Switchboard is being installed as a service.

For Linux: On Linux, to run Apache as a service, you need to create a startup script (usually in `/etc/rc.d` or a similar directory). The script should set the necessary DCF environment variables (these can be found in the `dcf.env` file in the Switchboard's installation directory) and export them. To run Apache, use the command "`run_apache.pl`". To stop it, use "`kill_apache.pl`". (See **Appendix F: Additional Information on Apache** for more information about running Apache as a service.)

For SUSE Linux 13.2: Note that the above commands may not work on SUSE Linux 13.2 and alternate commands may be required. For example, SUSE 13.2 uses a different Apache, started with

```
sudo /usr/sbin/httpd2
```

To start Apache automatically, you may need to put that command into the file `/etc/init.d/boot.local`

3.6 Uninstalling the DICOM Switchboard

Before uninstalling the DICOM Switchboard, please be sure that you have stopped it. This is done by opening the Switchboard Operations Window and clicking the "**Stop**" link. You should also stop the Apache server by closing the Switchboard Web Server command window that is running Apache, or stopping all instances of Apache via the Windows Task Manager, or by killing Apache by pressing CTRL-C in the command window that is running the Apache web server.

Uninstalling the DICOM Switchboard requires deleting the files that were installed and removing the environment settings required for the Switchboard's operation, as well as the shortcuts from the Start menu. These steps are automated in the **uninst-Switchboard.exe** program provided. (Remember that you must be logged in with Administrator privileges to install/uninstall the system and to adjust the system requirements.)

To uninstall:

- Click on the "*Uninstall Switchboard*" option in the DICOM Switchboard program group from the Start menu.
  (It may also be uninstalled via the Add/Remove Programs option on the Control Panel.)

  `Start → All Programs → Laurel Bridge Software → Switchboard → Uninstall Switchboard`

Note: If you have installed the DICOM Switchboard as a service, you must stop it and change it to a manually started service, via the Windows Control Panel and the Administrative Tools. See notes in Section 3.5.

For Linux:

To uninstall the DICOM Switchboard, make sure that its executables and servers are not running (the same commands for Windows shown above – e.g., stopping the Switchboard and then running `kill_apache.pl`) and then simply delete the Switchboard's installation directory. If you set the dcf_sysmgr to run as a service (as described above), you may need to kill that process before deleting the Switchboard.

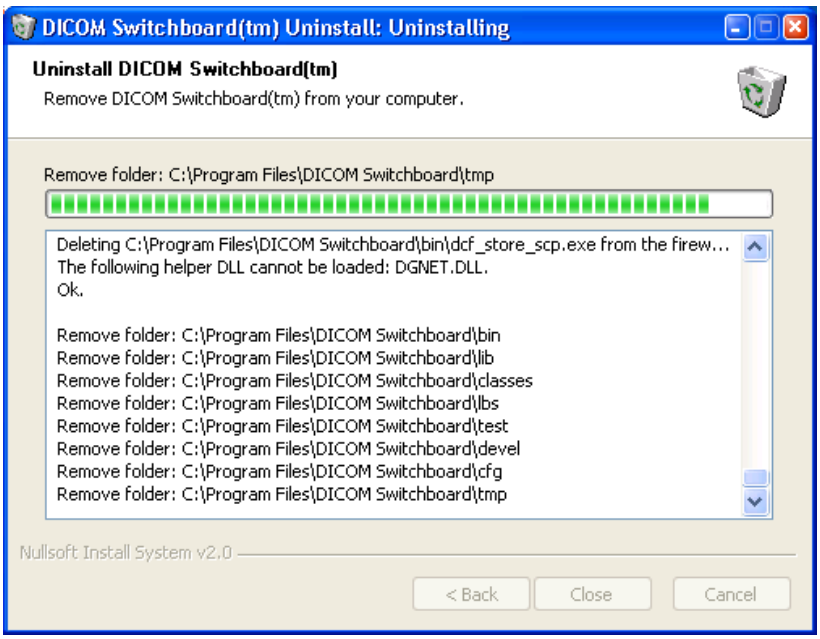


Figure 9 – Uninstalling the DICOM Switchboard

3.7 Upgrading to a newer Switchboard

If you have an older version of the DICOM Switchboard, you probably have many mapping rules and filters that you want to use when you upgrade to a new version of the Switchboard. But you don't want to have to enter all the mapping rules or filters again. As of Switchboard 3.5.0c, there is a script you can run to copy the configuration files from your old Switchboard directory to the new installation directory.

3.7.1 Upgrading on Windows:

Switchboard provides the script **update_switchboard.pl** to make it easy to migrate your Switchboard configuration. **Note** that this script should be run *before* you make any configuration changes to your new Switchboard, since those changes may be overwritten by the migrated changes.

1. Backup the `ext_data_dictionary` file (in the `cfg\dicom` directory) if you have modified it to a new location; you can copy it into the new installation if you need it.

2. Uninstall the old version of the Switchboard. The necessary Switchboard configuration files will not be deleted.

3. Install the new version of the DICOM Switchboard.

4. Open a **Switchboard Command Prompt** (for Linux, this is a Switchboard Linux Command Terminal).
   `Start → All Programs → Laurel Bridge Software → Switchboard → Switchboard Command Prompt`

5. Change to the **`bin`** directory.

6. Run this command: `perl update_switchboard.pl`

7. When prompted, enter the directory path of the old Switchboard installation.

8. If the script finds multiple files that could be migrated, you may be asked which one should be used. Answer the questions when prompted.

9. If the script succeeds, you will see a message like "*Success! Migration complete from…*". If you don't see that message when the script has completed, fix the errors and run the script again.

10. Once the script has succeeded, you will need to restart the Switchboard to use the migrated configuration – restart it via the web interface.

The script will copy over the configuration values from the old Switchboard, including any filters. Any private filters will be exported as global filter sets and will be referenced in the mapping rules.

Note: The migration utility will copy over *most* of the configuration changes that can be made through the **Switchboard's web interface**. If you *manually* edited the dcf_switch configuration file and changed some settings, you may need to manually edit the file and make those changes again.

Also, if you are migrating from Switchboard 3.3.46c or older and your filters used private tags, you should make sure that those tags are added to the *new* extended data dictionary – see **Appendix E: Editing the Extended Data Dictionary** for more information.


```
C:\Users\patrick\DICOM Switchboard 3.3.47a\bin>perl update_switchboard.pl

Enter the directory of the old Switchboard:  C:\users\patrick\Dicom Switchboard 3.3.42c
Old file: dcf_switch.2015.03.30-10.56.sav

Original configuration saved as: C:\USERS\patrick\DICOM Switchboard 3.3.47a\cfg/apps/defaults/dcf_switch-20150330-132958
.sav


Migrating from C++
Migrating to C++



There are multiple possible filter sets directories.
Please choose the correct one to use:
1: filter_sets.2015.03.30-10.56 - Copy
2: filter_sets.2015.03.30-10.56
2
Migrating filters from filter_sets.2015.03.30-10.56

Processing filter set: example_set
Copy filter set example_set as example_set_1
Processing rule EchoSCPRule
Processing rule StoreSCP1Rule
Processing rule EchoSCP2Rule
Processing rule FilterEchoSCPRule
        Filter set name: example_set
Processing rule JPEG70CompressRule
Copying group: default_group_1_2_3
Processing rule MoveFromSourceArchive
Copying group: migration_group_4
Rule set to use: default_group_1
Updating supported transfer syntaxes


Success! Migration complete from C:\users\patrick\Dicom Switchboard 3.3.42c

You must restart Switchboard to use the new configuration.

C:\Users\patrick\DICOM Switchboard 3.3.47a\bin>
```

Figure 10 – Migrating configuration to a new Switchboard

If you are moving your settings from a different machine, you should copy the Switchboard directory from the old machine to the new machine. Then run the `update_switchboard.pl` script as described above, specifying the directory that you copied when you are prompted.

3.7.2 Upgrading on Linux

1. Follow the standard Linux installation instructions to install Switchboard – see section **3.4 Installing the DICOM Switchboard on Linux** above.
2. "Dot" the dcf.env file to set the environment. (**`. ./dcf.env`**)
3. Activate the license as described in the instructions, or follow the steps described to install the license file that you previously downloaded.
4. From the new Switchboard's installation directory, run the script "`perl bin/update_switchboard.pl`". When prompted, enter the installation directory of the old Switchboard. The script will copy the configuration data from the old Switchboard to the new one.

For SUSE 13.2, you will need to do these *additional* steps to update any startup scripts; you will also need to update the Apache configuration to point to the new Switchboard.

5. Copy the `switchboard_ctl.sh` startup script from the old Switchboard directory to the new one, if it doesn't already exist.
6. Edit the file `/etc/systemd/system/switchboard.service` and change the old version to the new version.
7. As root, update the services by running "`systemctl daemon-reload`".
8. As root, edit the `/etc/apache2/default-server.conf` file and change references from the old version to the new version. You will need to restart Apache after this.
9. You can now use the `systemctl` commands to stop and restart Switchboard – these commands are described in section **3.5.1 Configuring the Switchboard service on Linux** above.

3.8 Duplicating the configuration

If you are retiring one machine and moving the settings from the old server to a new server but don't want to reenter all your settings, you will need to copy these files – copy them from the installation directory on the old server to the corresponding directory on the new server:

```
cfg/apps/defaults/dcf_echo_scp
cfg/apps/defaults/dcf_store_scp
cfg/apps/defaults/dcf_switch
cfg/apps/defaults/DLOG_Server
cfg/apps/defaults/RTLog_Server
cfg/apps/defaults/DCDS_Server
cfg/dicom/ae_title_mappings
cfg/dicom/ext_data_dictionary
All the files in cfg/dicom/filter_sets
All the files in cfg/dicom/session_settings
```

The bolded files are required – these are your Mapping Rules, Filters, and related files; the others can be easily configured via the UI. If the installation directory is different on the new machine, there may be minor path differences for directories – these can be easily edited in VIM or Notepad. You will have to restart Switchboard to use the new configuration settings.

If you want to have multiple Switchboards running but all using the same settings, you should read the section on **Synchronizing Multiple Switchboards**.

4 Running the DICOM Switchboard

4.1 Starting the DICOM Switchboard

A web-based interface is used to control and to configure the DICOM Switchboard. A web server runs on the Switchboard's host system, allowing both local and remote access to the Switchboard configuration and monitoring utilities. The web server must first be started in order to operate the DICOM Switchboard. The steps to start the web server and access the Switchboards' web interface are described here.

For Windows:

From the Windows *Start* menu, select "Laurel Bridge Software", "DICOM Switchboard" and then "Start Switchboard". This starts an Apache web server, then starts the Switchboard's servers (if these are not already started), and, finally, opens a web browser that points to the Switchboard Operations Window (described below in Chapter 5).

`Start → All Programs → Laurel Bridge Software → Switchboard → Start Switchboard`

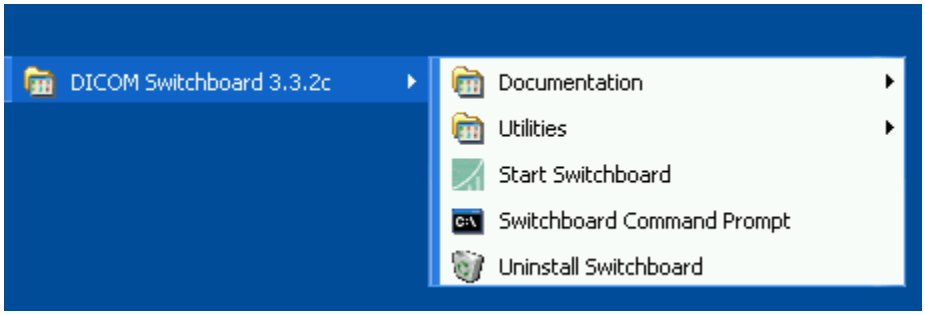


Figure 11 – Start Menu option for the DICOM Switchboard

Note: If the web browser opens but the Switchboard Operations window reports that the Switchboard will not start, there may be an issue with the firewall and hostname and/or ports in use – see Sections 14.1.3 and 14.1.4 for help in resolving those specific issues. (**This applies to both Windows and Linux**.)

For Linux:

Open a command terminal and "cd" to the directory where you installed the DICOM Switchboard. Set the environment by "dotting" the dcf.env environment file ("`. ./dcf.env`"). (Recall that this is referred to as a "*Switchboard Linux Command Terminal*".)

Start the Apache web server for the DICOM Switchboard by running the command `run_apache.pl`.

Open a browser window and point it to the machine hosting the DICOM Switchboard and the Apache port you specified when you set up the Switchboard, e.g., `http://mymachine:8081`.

Click the **Start** link to start the DICOM Switchboard (and its servers), if it is not already started.

Note that SUSE 13.2 uses different commands, detailed in Section **3.5.1 Configuring the Switchboard service on Linux** above.

Running the Switchboard without the Web server on Linux:

You may want to run the Switchboard without the web server – let's say that you have it configured exactly as you want it, and you want the Switchboard – but not Apache – to start automatically when the server boots up. To do this, you will need to add the DCF environment variables (these can be seen in the `dcf.env` file in the Switchboard's installation directory) to your environment or startup script – be sure that their values are being properly exported so that they can be used. Use the following commands to start and stop the Switchboard:

To start the Switchboard:

```
$DCF_ROOT/bin/dcfrestart.pl -c $DCF_CFG/systems/dsb_switch_unix.cfg
```

To stop the Switchboard:

```
$DCF_ROOT/bin/dcfstop.pl
```

Note: DCF_ROOT is the directory where you installed the Switchboard. Depending on your PATH settings, you may be able to omit "$DCF_ROOT/bin" from those commands, or you can hard-code the paths to the scripts if you so desire.

These commands (and their equivalents on Windows) can also be used to start and stop the Switchboard manually, without using the web interface.

Note that SUSE 13.2 uses different commands, detailed in Section **3.5.1 Configuring the Switchboard service on Linux** above.

4.2 Verifying the DICOM Switchboard's Operation

Now that you have installed the DICOM Switchboard and started it, you may wish to make sure that it is operating correctly. The DICOM Switchboard comes "out of the box" with a configuration that makes it simple to test that it is working – it includes mapping rules that allow you to perform an Echo test (or "ping") to a built-in DICOM Verification (Echo) Server without making any changes to the Switchboard configuration.

1. From the Switchboard Operations Window, click the link "*View Status Monitor*". This will open a monitoring app in a new window.

2. From the Windows Start menu, launch the Switchboard's Echo Client.

   `Start → All Programs → Laurel Bridge Software → Switchboard → Utilities → Run Echo Client`

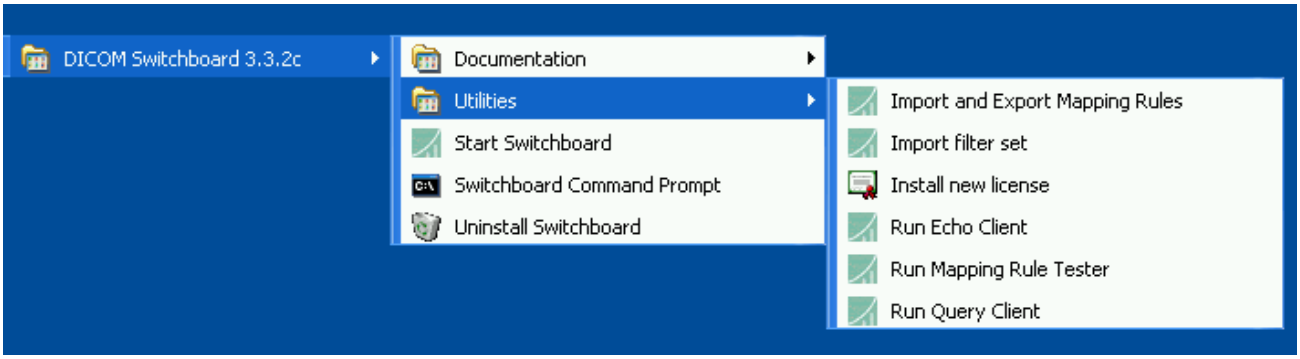


Figure 12 – DICOM Switchboard utilities menu

For Linux: From a Switchboard Linux Command Terminal, launch the Echo Client via:

```
<instdir>/bin/JEcho_SCU.bash
```

3. Click the "*Send DICOM Echo*" button, and watch the Status Monitor. You should briefly see a green line of data, indicating that the Echo request is going through the DICOM Switchboard (to the Switchboard's Echo SCP). Also, the Echo client should show "Success" in its Echo Status box.
4. Your DICOM Switchboard is working! The rest of this manual will guide you on configuring the DICOM Switchboard for your own custom monitoring, routing, and filtering needs.

(Section 12.1 has more information on this test and on using the Echo SCU and SCP; see Section 12.4 for information on testing and using the Store SCU and SCP.)

5 Switchboard Operations Window

This is the screen that appears when you start the DICOM Switchboard from the Windows Start menu or from the desktop shortcut (if you chose to create one during installation).

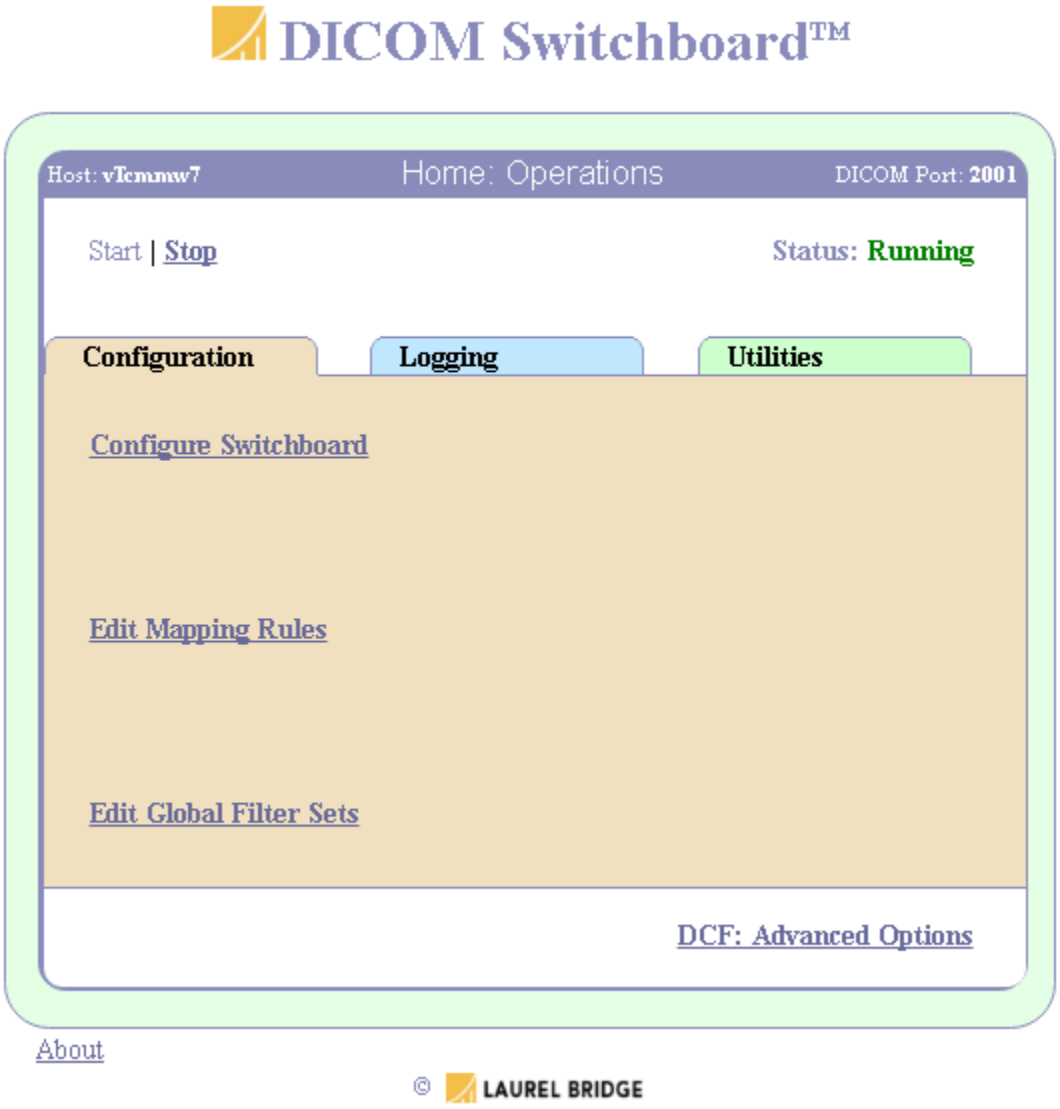


Figure 13 – DICOM Switchboard: Main Operations Window

This window is the central control point for the DICOM Switchboard and its operations. From here you can select any of the options for configuring the DICOM Switchboard, editing its Mapping Rules and Filter Sets, operating it, or using its utilities.

This window also provides a snapshot of the status of the DICOM Switchboard and important information about its operation. Its status – "Running" in the example above– can be seen in the upper right. The status is also shown in the light green frame around the window; this will be light red if the Switchboard is stopped. You can start and stop the Switchboard from this page. The host and port for the Switchboard are in the upper left and upper right corners, respectively, making it easy to know how your modalities should be reconfigured to communicate with the DICOM Switchboard.

From this window you can select the Configuration, Logging, or Utilities tabs to access each of their options.

Figure 14 – DICOM Switchboard: Operations Tabs

The Configuration tab has options such as setting up the DICOM Switchboard's filters and the mapping rules. The Logging tab lets you view the log files, access the Real-Time Log, and adjust the logging verbosity (to help you debug problems). The Utilities tab lets you view the Status Monitor, as well as letting you use web interfaces for other utilities.

The DICOM Switchboard is built on the Laurel Bridge Software DICOM Connectivity Framework (DCF). There are many more advanced ways to configure and modify the Switchboard's behavior, including more debugging flags that can be enabled or more configuration options to select. Access these by clicking on the "DCF: Advanced Options" link. These options are designed for experienced users of the DCF and should be used with caution, but they are available should you need them.

Figure 15 – DCF: Advanced Operations

For example, if you click on the link "Configure dsb_switch_win32.cfg" (as seen above in **Figure 15**), you can also adjust more attributes than those shown below in Chapter **7, Configuring the DICOM Switchboard**, additional settings including the maximum sizes of the log files and timeout values for the servers.

6 Step-by-Step Setup Guide

This section will guide you through each step of starting the DICOM Switchboard. These are the basic steps, so you should consult the appropriate chapters of the manual for more detail on each of these steps and the options available to you.

1. Start the system as described in Section **4.1 Starting the DICOM Switchboard** above.
2. You may wish to clear the log files before monitoring/filtering any connections; do this by clicking the "Clear Log Files" link on the Logging tab. (This will make it easier to interpret the data by removing any old connection information.)
3. Configure the Switchboard's ports and other settings by selecting the "Configure Switchboard" link from the "Configurations" tab. (These and additional settings may also be accessed by selecting "DCF: Advanced Options" and then the "Configure dsb_switch_win32.cfg" option [or the "Configure dsb_switch_unix.cfg" option if you are running on **Linux**].) You may need to modify your firewall configuration to allow communication on the selected port (and on the web server's selected port) and/or by the Switchboard's applications. (See Chapter **7 Configuring the DICOM Switchboard** for more information.)
4. Start the Filter Set Editor by clicking the "Edit Global Filter Sets" link. This will open up a web application in a new window. You may use the app to view and edit the sets of *global* filters used by the DICOM Switchboard. (See Chapter **14 DICOM Switchboard Applet Issues** for information about the applets used by the DICOM Switchboard.)
5. Create the filters to be used by the DICOM Switchboard. You may create new sets of shared (or global) filters and add filters to them; the filters to use will be selected when you set up the mapping rules below. (Global filters are a convenience for when you need to attach the same filters to multiple mapping rules.)

   Filters may be created to remove elements, to replace the value of an element, to modify an element's value with regular expressions, or even to modify the pixel data itself by bit-shifting it. (See Section **8.2 Creating Filters and Filter Sets** for more information on how to use the Filter Editor or for more detail on the types of filters built into the Switchboard. There is a brief guide to the filters at the beginning of Chapter **8 Filters and Filter Sets**.) **Note** that if your filters are operating on private tags/elements (VR: UN), you should add the tags to the "Extended Data Dictionary" so that the content of the element is correctly interpreted – this is described in **Appendix E: Editing the Extended Data Dictionary**.

   If you need many similar filters (e.g., hundreds or thousands), see **Appendix H: Using & Creating Many Mapping Rules** for information on how to use scripts to create them in batch mode.

   When you are done, exit the Filter Set Editor by closing the window.
6. Start the Switchboard's Mapping Rule Editor by clicking the "Edit Mapping Rules" link. This will open up a web application in a new window. You may use the app to view and edit the mapping rules used by the DICOM Switchboard.
7. Add new mapping rules or edit the existing mapping rules as necessary for your systems. You will have to modify the rules in order to indicate what AEs are handled by the DICOM Switchboard and where the data for such AEs is sent. **Note** that any changes to the mapping rules will not take effect until you "Save and Apply" them – this will commit the changes to disk and load them into the Switchboard.

If any of the rules use "Filtered Mode", you should attach filters to each rule that needs to filter the data – the global filters are edited through the Filter Set Editor. Select the filters that you wish to use.

See Section **9.2 Mapping Rule Editor** for help on using the Rule Editor; see Section **9.1 About Mapping Rules** for information on how the mapping rules work. The examples in Section **9.2.7** can help you understand how to configure your mapping rules.

If you need many similar mapping rules (e.g., hundreds or thousands), see **Appendix H: Using & Creating Many Mapping Rules** for information on how to use scripts to create them in batch mode.

8. Once you have made all the changes you wish, exit the Mapping Rule Editor by closing the window. (**Remember** to "Save and Apply" your changes.)
9. You may adjust the amount of information about the connections that is logged by clicking on "Set Logging Verbosity" from the Logging tab and selecting the data that you wish to have logged. (The fields are – more or less – increasing in verbosity from top to bottom.) Click "Save these settings for next time" if the logging changes should be used the next time the DICOM Switchboard is started. (Note that logging changes do *not* require the Switchboard to be restarted.)
10. The DICOM Switchboard **must** be restarted before the new mapping rules and configuration settings will take effect, so restart it by clicking "Stop" on the Switchboard Operations page. You are returned to the Switchboard Operations page after the system has stopped. Once it has stopped, restart the DICOM Switchboard by clicking "Start" on the Switchboard Operations page.
11. Configure any clients that are to use the DICOM Switchboard, by changing the host and port that they are sending to. (The Switchboard's port information was set by you back in step #**3** using the "Configure Switchboard" page. Remember that you may need to adjust your firewall's configuration for the new ports and for the Switchboard's apps.) Once your clients are configured, proceed with your tests. (See **6.1 Monitoring / Filtering an Association** below for more information on configuring a connection to use the DICOM Switchboard.)
12. Configure the destinations as necessary to receive from the DICOM Switchboard. This may include setting them to recognize communications from the Switchboard's host computer.
13. You may observe the state of the connections through the DICOM Switchboard by launching the Status Monitor app; this is done by clicking the "View Status Monitor" link on the Utilities tab of the Switchboard Operations page. This app displays information about the current connections to the DICOM Switchboard. If the "lights" are not flashing, it means that there is no activity through the DICOM Switchboard; you may want to make sure that your mapping rules are correct and that your clients (configured in the previous step) are sending to the DICOM Switchboard.
14. Another way to verify that the DICOM Switchboard is operating correctly is to view the log files generated by it. Every connection that goes through the Switchboard's Router generates a log message that displays its information, both incoming (into the Switchboard) and outgoing (on to its destination). Clicking the "View Log Files" link on the Logging tab will take you to a page displaying all the log files generated by the DICOM Switchboard; here you may select a log file to view in your browser. The "system.log" and the log file for the "dcf_switch" will contain information on transactions being processed by the Switchboard. A simple way to check that the Switchboard is operating correctly is to send an "echo" through

the DICOM Switchboard and then check the log files to see that the passage of the data is logged.

6.1 Monitoring / Filtering an Association

The DICOM Switchboard must be interposed in a connection in order to monitor it and filter the data: SCUs send their data to the DICOM Switchboard, which sends it on to the SCPs.

1. Using the Mapping Rule Editor, create a mapping rule that will take data matching the desired AE and send it on to the appropriate SCP. Make sure you have the "host" and "port" fields for the SCP set correctly. (Recall that some changes to the mapping rules may require the DICOM Switchboard to be restarted.)
2. Configure the SCU so that it sends its data to the DICOM Switchboard instead of directly to the SCP. Primarily this means changing the destination address and port of the SCU to be the host and port of the DICOM Switchboard.
3. If necessary, configure the SCP to accept connections from the Switchboard. This could mean, for example, configuring it to accept messages from the Switchboard's address or to recognize the new AE titles from the Switchboard.
4. Run your connections and tests. You may find that adjusting the logging verbosity will make it easier to determine any problems that may exist in the connection.

Note: As you test your Switchboard's configuration, you may wish to clear your destination SCP's database or cache periodically. This is because users often test the Switchboard by sending the same image multiple times. In many cases, the receiving SCP will correctly receive the image the first time; after that, it will think that it already has the image and so report success but it will not have actually received the image – it could report success but just throw the incoming image away and use the previously cached data. If this happens, you may not see the desired AE title or filter changes. Clearing the destination's database before testing the Switchboard can prevent this kind of problem.

7 Configuring the DICOM Switchboard

The Configuration screen allows the user to set the basic DICOM communications settings for the application. It is particularly important that you note the TCP Port being used – this is the port that devices will use when they send DICOM data to the DICOM Switchboard.

Figure 16 – DICOM Switchboard: Configuration Window

When you are configuring the DICOM Switchboard for the first time, you will probably want to make the following modifications to the configuration:

- Set the DICOM / TCP port for the Switchboard to use. See the DICOM standard for additional information on TCP ports (PS 3.8-2007 Network Communication Support, section 9.1.1). Typically ports 104 or 11112 (decimal) may be used by DICOM applications. (Remember that the port selected must not be used by any other applications. Also, your firewall configuration may need to be modified to allow communication on the selected port.)
- For multi-homed machines (such as those on a VPN), set the Server Host Address fields for each server. This is so that the DICOM Switchboard knows which address to listen to.
- Set the maximum number of concurrent associations that you want to accept. The maximum theoretical limit is 1024, but the actual limit depends on your hardware.
- Set the Socket Port for the DLOG Server. You may also need to set the Log Server's Host Address field. (The DLOG Server is used by the DICOM Switchboard to write the log files and is instrumental for the Real-Time Log.)
- Set the Port for the Real-Time Logger. (This is not the same as the Socket Port for the DLOG Server above.) This is used by the Real-Time Log Viewer web application's scripts to retrieve log data when needed.
- You may need to set the Config Server's Host Address field for the DCDS Server. (The DCDS Server is used by the DICOM Switchboard to manage the Switchboard's configuration data.) This helps the Mapping Rule Editor and other Java apps and applets to communicate with the DCDS Server.
- Set the TCP port for the Switchboard's Echo Server.
- Set the TCP port for the Switchboard's Store Server if you will be using it for testing purposes.
- Adjust the other configuration attributes as needed. (It is not recommended that you disable statistics for the DICOM Switchboard – doing so will result in no data being collected for the Status Monitor: it will not show any data and may thus appear to be broken.)

After all the fields have been filled in to your satisfaction, click the "*Update*" button to save the new values. (Please note that changes to the configuration values require the DICOM Switchboard to be restarted before they will take effect. Restart the DICOM Switchboard by clicking "*Stop*" on the Operations Window, then return to the Operations Window and click "*Start*".)

Note: On some Windows systems, conflicts may occur between the ports used by the operating system and the ports used by the DICOM Switchboard. Specifically, the default ports of the Switchboard's servers may be used by Windows **svchost.exe**. If this problem occurs, you should change the default ports used by the Switchboard's servers. This can be done via the Switchboard Configuration window (shown **above**); if you need to change more settings (such as timeout values or log file settings), follow the Advanced Steps below.

Advanced Steps:

a. Start the DICOM Switchboard via the web browser by clicking the "*Start*" link.

b. Return to the Switchboard Operations Window, and select "*DCF: Advanced Options*."

c. Select "*Configure dsb_switch_win32.cfg*".

d. Change the TCP Port setting for the servers. (Please be sure that none of the port values conflicts with any other.)

e. Click "*Update*" to save the changed values.

f. Click "*Home*" to return to the Switchboard Operations Window.

g. You may need to update the mapping rules (via the "*Edit Mapping Rules*" link) if the DICOM Switchboard is routing to servers on the same machine.

h. Restart the DICOM Switchboard by clicking "*Stop*", then return to the Switchboard Operations Window and click "*Start*".

7.1 Synchronizing Multiple Switchboards

If you have more than one Switchboard installed and operating, you may want them to share the same filters and the same mapping rules, but you probably don't want to repeat the process of entering the data on multiple systems. The Switchboard allows you to set one Switchboard installation to act as the "Master", and then other Switchboard installations can be marked as "Slave" systems. If configured in this way, then, when you update the mapping rules or filters on the master box, the slave boxes automatically synchronize their data to the Master.

Any copy of Switchboard may be configured as a Slave or a Master; the default operating mode is Master, which allows that installation to fully manage its own configuration. Please note that the Apache web server for the master Switchboard **must** be running in order for the automatic synchronization to work.

7.1.1 Slave Configuration

To configure a Switchboard installation as Slave to a Master, from the Switchboard Operations Window:

- Select the Configuration tab, and then click "Configure Switchboard".

  Under the section "DCDS (Configuration Data) Server Configuration", click the checkbox next to "Run as slave".

  (Leave the box unchecked for the Master system.)

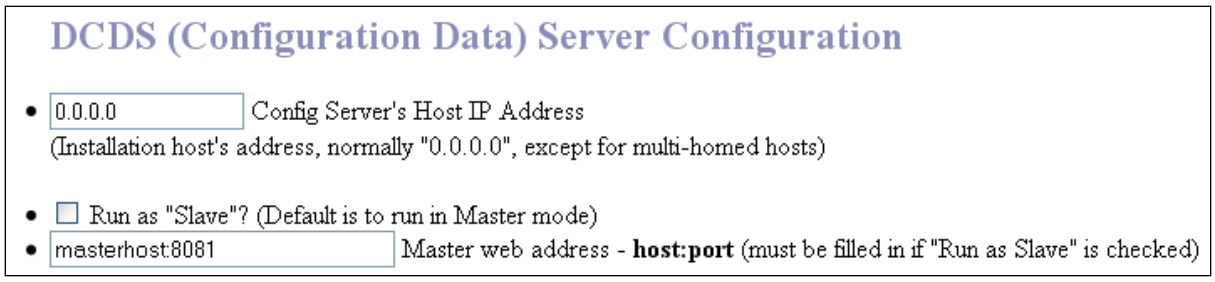


Figure 17 – Configuring a Switchboard as a "slave"

- Fill in the field – "Master web address" – with the host and port of the **web server** running and serving the Switchboard on the master box, e.g., "master.company.com:8081".
  Notes:
  - Do **not** put "http://" at the beginning of the system address.
  - The port (8081) **above** is the one selected and set for the web server during the original software installation for the Master system. Do **not** use the DICOM port for the Master.
- Save the configuration changes. After saving, you will have to restart the Switchboard in order to have it operate as a slave to the designated master.

When one Switchboard is slaved to another, any changes to the mapping rules, filter sets, or debug flags on the Master system will be automatically copied over to the Slave. Therefore, when Switchboard is operating in Slave configuration, the editors for the mapping rules, filters, and debug flags are disabled. (See Section **11.1 Debugging a Switchboard in Slave mode** for how to debug an individual Slave box.)

When operating in Slave mode the Switchboard will display a message on the Configuration tab screen that indicates the Master system: "`Synced with master at <hostname>:<port>`".

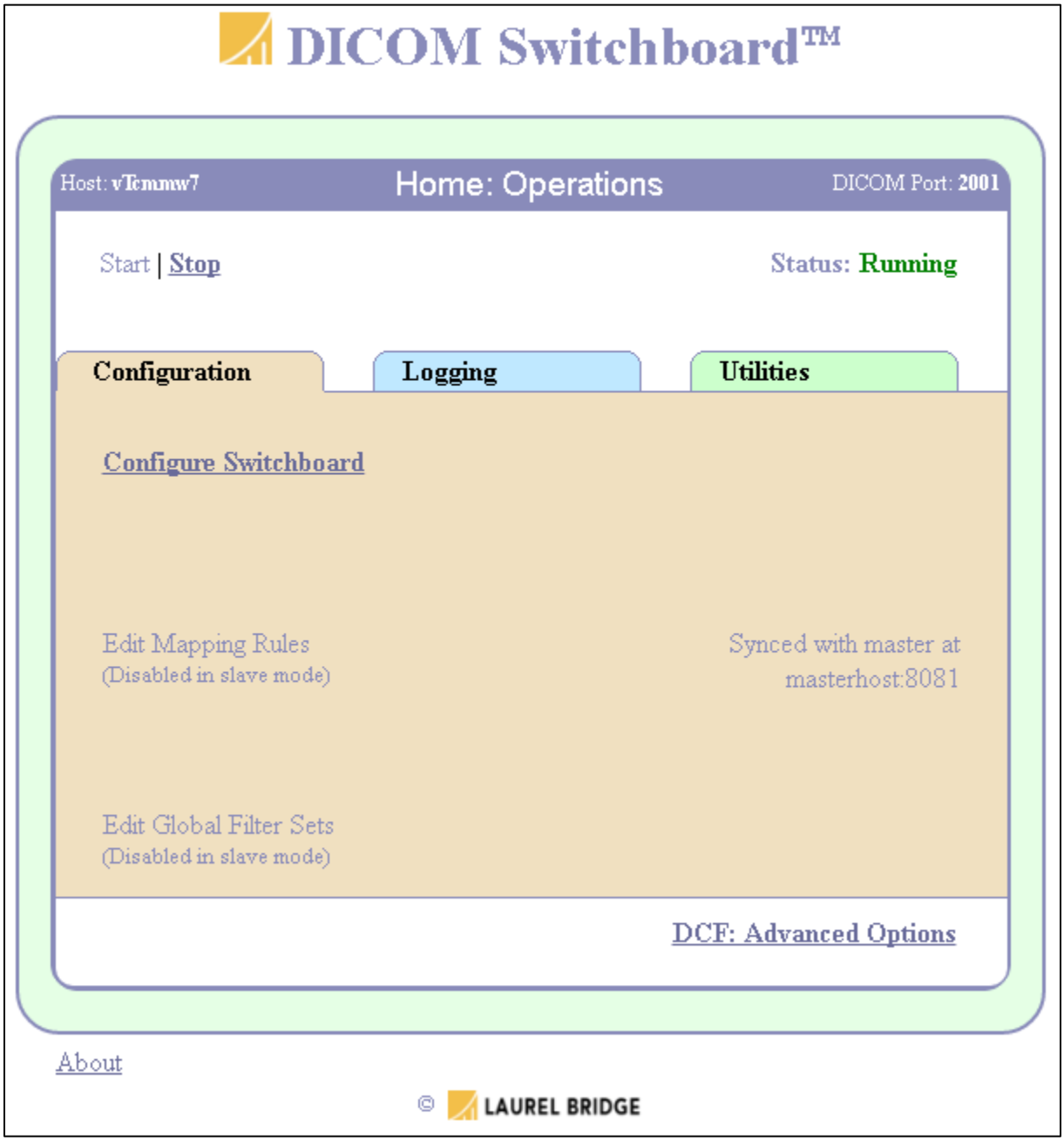


Figure 18 – A Switchboard running in "slave" mode

8 Filters and Filter Sets

The DICOM Switchboard can filter the data contained in DICOM messages as those messages are transmitted between devices – modifying DICOM elements, adding elements, changing the values, or even deleting elements. This capability can be useful for correcting data before it is received by the destination. In fact, this is one of the primary purposes of the DICOM Switchboard: to modify data that is being routed through it.

There are many different filters that are available in the DICOM Switchboard, giving it great power to do almost any filtering operation you want. The table below can help you figure out which filter you want to use.

If you want to:	Use this Filter:
• Remove elements from a dataset	The **Remove** filter lets you remove only those tags that you choose. (See Section **8.4** below.) The **Copy** filter allows you to remove many tags at once by only allowing specified tags to be passed (copied) through. (See Section **8.4** below.)
• Add a new tag and value	The **Add/Replace** filter lets you specify an existing or missing tag and the value it should have. (See Section **8.4**.)
• Change the value of a tag	The **Add/Replace** filter can change the value of an existing tag to be a new value. (See Section **8.4** below.) The **Modify** filter can also change the value of an existing tag. (See Section **8.4.**)
• Modify the value of a tag via regular expressions	The **Modify** filter can use regular expressions to parse a tag's value and rearrange or replace the parts. (See Section **8.4**.)
• Use regular expressions to combine the values of multiple tags or to split one tag into multiple tags	The **Composer** filter can use regular expressions to parse the values of several tags and then rearrange the parts. (See Section **8.4.2**.)
• Change the case of a tag's value	The **Modify** filter can change the case of a value – the result can be all UPPER-CASE or all lower-case. (See Section **8.4**.)
• Copy or move the value of a tag to a new tag	The **Modify** filter can copy the value of one tag to another tag. (See Section **8.4**.)
• Pad a value to be a desired length	The **Pad Value** filter lets you pad a value with leading or trailing characters. (See Section **8.6**.)
• Modify the pixel data	The **Pixel Shift Bits** filter can shift the bits in the pixel data of an image. (See Section **8.5**.)
• Convert a color image from Interleaved to Planar or vice versa	The **Planar Configuration Convert** filter can convert an image from Interleaved format to Planar format or vice versa. (See Section **8.5.1**.)
• Change the values of several tags based on the values of one or more tags	The **Mapping List** filter is designed to check the value of a tag(s) and, based on that value, choose the new values for other tags from a list. (See Section **8.4.1**.)
• Fix a bad Icon Image Sequence	The **Fix Icon Image Sequence** filter will try to read the data; if it can't be read, the bad sequence will be deleted. (See Section **8.5.2**.)
• Drop a message entirely or change a Request to a Response	See **Appendix J: Specialized Filtering**.

Figure 19 – Guide to Filters

These filters are described in more detail below; the examples (Section **8.8** below) can also help, and **Appendix A: Example Filter: Modifying Elements** has more information on how to use the Modify filter. The Switchboard also has advanced Conditional Filters – these are described in **Appendix I: Conditional Filters**.

You can use many filters in conjunction with each other to do a great many things. For example, because the filters are applied sequentially, your first filter could change a value (Add/Replace), your second filter could copy that value to a new tag (Modify), and a third filter could – based on that new value – change the values of several other tags (Mapping List).

Note that most of the filter types included with the DICOM Switchboard are designed to operate on *string* values. To ensure that the values of private tags/elements (VR: UN) are processed correctly, they should be added to the Extended Data Dictionary (this process is described in **Appendix E: Editing the Extended Data Dictionary**). If you do not do this, then it is possible that the private element would be interpreted as a stringified binary value (VR: UN), which would cause the processing of the values to be incorrect and the results may be unpredictable.

Note on Custom Filters: Switchboard versions 3.5.0c and higher can use custom filter classes written in Java to do more advanced filtering than can be defined in the Filter Editor GUI. Custom filter configurations must be added *manually* (via a text editor such at Notepad or GVIM) to the filters defined in Switchboard's `cfg\dicom\filter_sets` subdirectory, while the Java classes themselves should be added appropriately to Switchboard's `classes` directory – for example, filters written by Laurel Bridge Software developers usually go in the directory `classes\com\lbs\CustomFilters`. The filter configuration should have the name for the filter and the name of the filter's Java class, as well as any parameters that the filter needs. In the example custom filter configuration below, the filter's name is "example 1", the **filter_class_name** attribute has the name of the Java class (com.lbs.CustomFilters.MySQElementFilter), and the parameters **sq_to_check** and **sq_for_saved_copy** are read and used by the filter class in its processing of the data.

```
[ example 1 ]
filter_class_name = com.lbs.CustomFilters.MySQElementFilter
sq_to_check = 0040,0260
sq_for_saved_copy = 0077,1000
```

8.1 Filter Sets

From the Switchboard Operations Window, you can start the Switchboard's Filter Set Editor web application by clicking the "*Edit Global Filter Sets*" link (on the Configuration tab). This will open up a new window that looks like the one below:

Figure 20 – A view of the Filter Set Editor

Note: As of version 3.3.40c, the Switchboard uses a non-applet version of the Filter Set Editor that is more browser-friendly. However, the old Java version is still available and may be used in a non-applet mode.

> **For Linux:** The Filter Set Editor (Java app) may also be launched from a Switchboard Linux Command Terminal by running the command "`bin/Run_FilterSetEditor.bash`".
>
> Note: On Windows, the Filter Set Editor (Java app) can be run via the command "`Run_FilterSetEditor.bat`" from a Switchboard Command Prompt.

The Switchboard's Filter Set Editor allows you to edit global (shared) filter sets and their associated filters to configure how the DICOM messages being processed should be modified.

Consider the following example scenarios:

- Suppose that the source has the patient name (tag: 0010,0010) in an incorrect format, such as first name first, instead of last name first. You could set up a filter that used regular expressions to reverse the name value ordering to the correct ordering.
- Suppose that the destination expects some data to be contained in element "B" instead of in element "A". You could set up a filter to copy the data from element "A" to the desired DICOM element, "B". Element "A" could be left as is or deleted by a subsequent filter rule.
- Suppose you do not want certain elements in the source datasets to be included as they are sent to the destination. You could set up filters to screen those particular unwanted elements from the datasets before they are forwarded to the destination.

The DICOM Switchboard can do all this and more, allowing your DICOM communication to proceed smoothly and without requiring user interaction to correct the data on the destination host after the data arrives.

Note: see the additional information that covers the Modify filter options in **Appendix A: Example Filter: Modifying Elements**.

The filters you create here can be used when you set up the device's Mapping Rules by associating a filter set with a Mapping Rule. When you are done creating filters and have saved them, simply close the Editor by closing its window.

8.2 Creating Filters and Filter Sets

The DICOM Switchboard and its filtering capabilities allow you to change the data in DICOM messages – modifying DICOM elements, adding elements, changing the values, or even deleting them. The filters created are grouped into "Filter Sets" so that they can be shared by many applications and attached to multiple mapping rules.

When you use the Filter Set Editor, you will see a table with the list of global filter sets (these are usually located in the local file system in `$DCF_CFG\dicom\filter_sets`). Buttons below the table allow you to Add, Edit, Delete, Copy, or Rename filter sets. (See the example window in **Figure 20** above.)

- Clicking the "*Add Set*" button will prompt you for the name of the new filter set. (Note that the name must be unique and that some special characters – such as quotes and slashes – are not allowed.) Once you have added a filter set you should edit the new set to add or define the filters to be included in that set.
- To edit a filter set, select it in the table and then click the "*Edit Set*" button. The screen will change to a table with a list of the filters in the set; see Section **8.3** below for information on editing a particular filter.
- To delete a set, select it in the table and click the "*Delete Set*" button. You will be asked to confirm the deletion before the filter set is permanently removed.
- To copy a set, select it in the table and click the "*Copy Set*" button. You will be prompted for the name of the new filter set.
- To rename a set, select it in the table and click the "*Rename Set*" button. You will be prompted for the new name for the filter set.

Note that you may have to restart the DICOM Switchboard before the filter changes can take effect. And once you have created a global filter set, it will still need to be attached to a mapping rule in order for data passing through the Switchboard to be filtered.

8.3 Editing the Filters

Once you have selected the filter set to edit, you will see a list of the filters in the set. Buttons on the right side of the table allow you to Add, Edit, Delete, Copy, or Rename particular filters; arrow buttons on the left side of the table allow you to change the order of the filters.

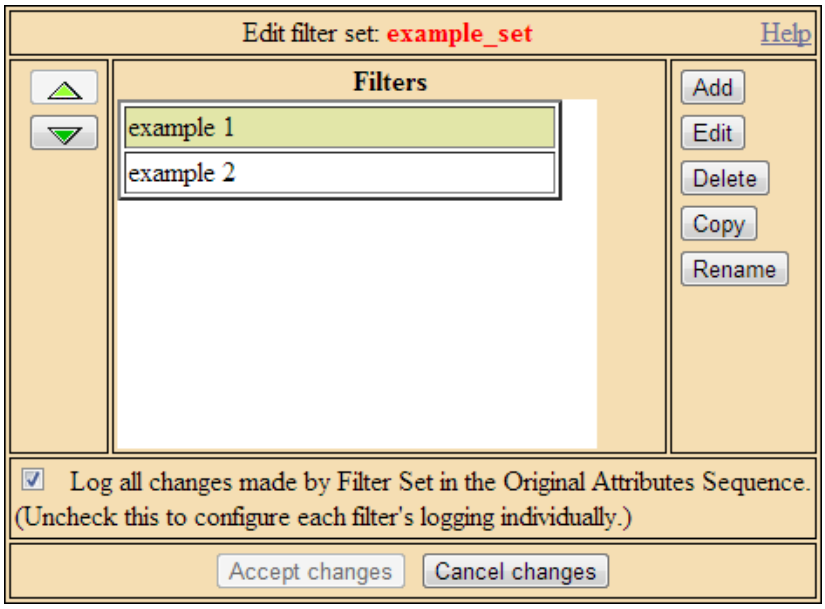


Figure 21 – A view of the Filter Editor

For example, **Figure 21** shows the list of filter rules in the filter set called "example_set".

- Clicking the "*Add*" button will switch to a screen where you can enter the name of the new filter and select its type from a drop-down list (see below). The new filter will automatically be opened for editing.

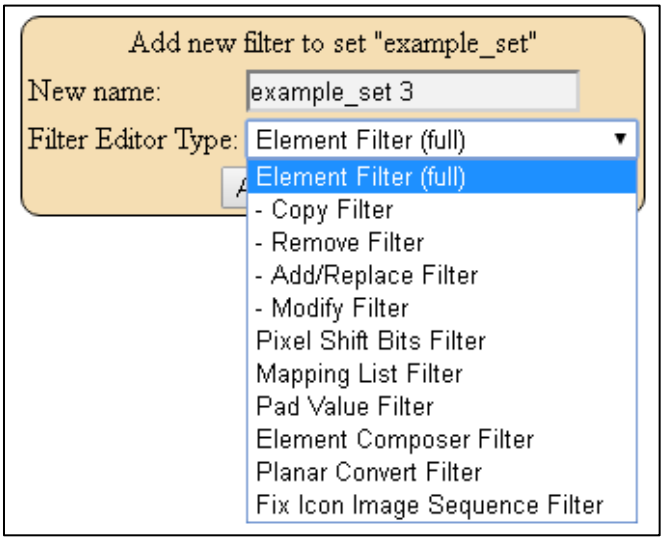


Figure 22 – Selecting a filter type in the Switchboard

There are many filter types and operations possible with each filter; individual filters are not described here. (The various types of Element Filters are described below [in Section **8.4**], as is the usage of the Pixel Shift Bits Filter and the other filters.)

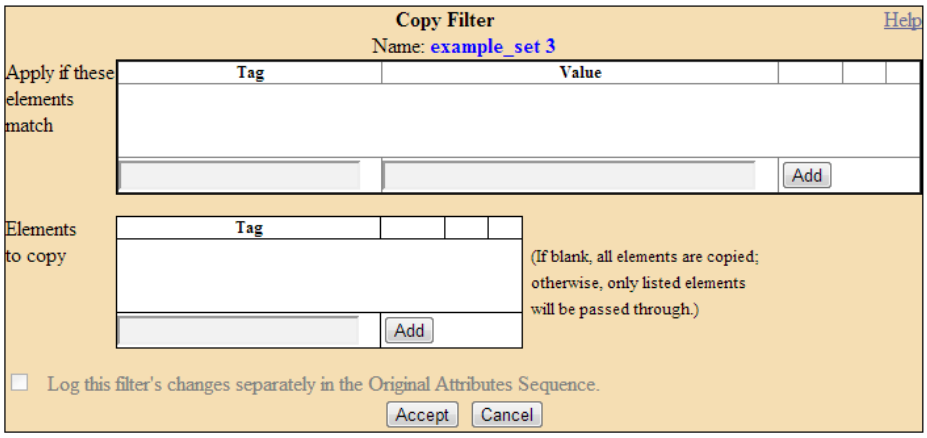
- To edit a particular filter, select it in the table and click the "*Edit*" button. This will open a new window with the filter's data displayed in a panel with fields appropriate to the filter's type and its purpose.
- To delete a particular filter, select it in the table and click the "*Delete*" button. You will be asked to confirm the deletion before the filter is permanently removed.
- To copy a particular filter, select it in the table and click the "*Copy*" button. You will be prompted for the name of the new filter.
- To rename a particular filter, selected it in the table and click the "*Rename*" button. You will be prompted for the new name for the filter.
- The filters are applied in the order that they are specified. There may be times when you want a filter applied before another filter, but it occurs later in the list. You can use the arrows on the left side of the list to move the filters up and down in the list; to do this, select the filter you wish to move by clicking on it with the mouse pointer, then click on the appropriate arrow key to move the filter up or down in the list.
- You may also specify if the changes made by the filters should be logged to the Original Attributes Sequence (OAS) – this can be used to provide a "history" of how the data was filtered. When this is selected at the **individual filter** level, a separate entry is added to the OAS for each filter's changes. If this option is set at the **filter set** level, then just one entry is added to the OAS for all changes made by the filter set.
- Once you are done modifying the selected filter set, click "*Accept changes*" to save the changes you have made, or "*Cancel changes"* to discard them. See **Figure 21** above.

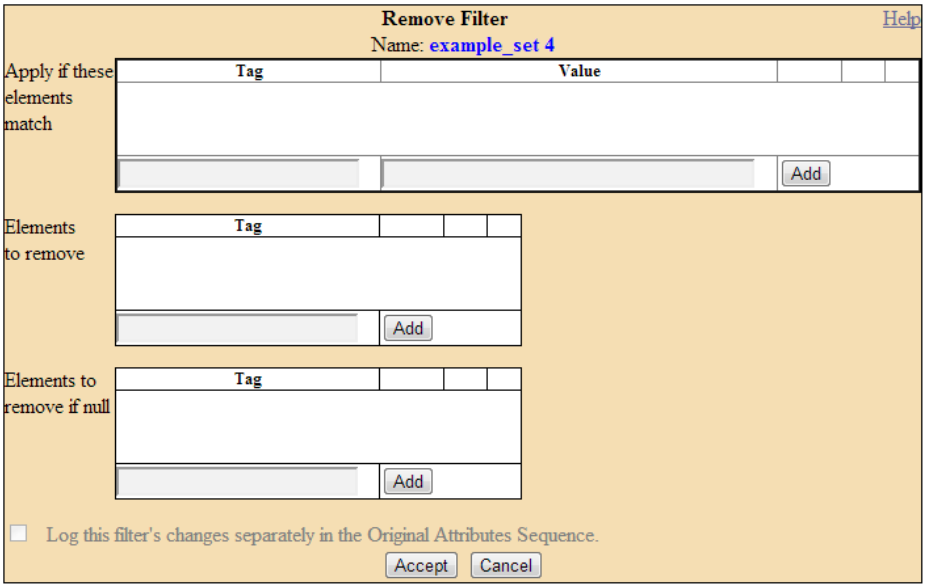
8.4 Filtering DICOM Elements

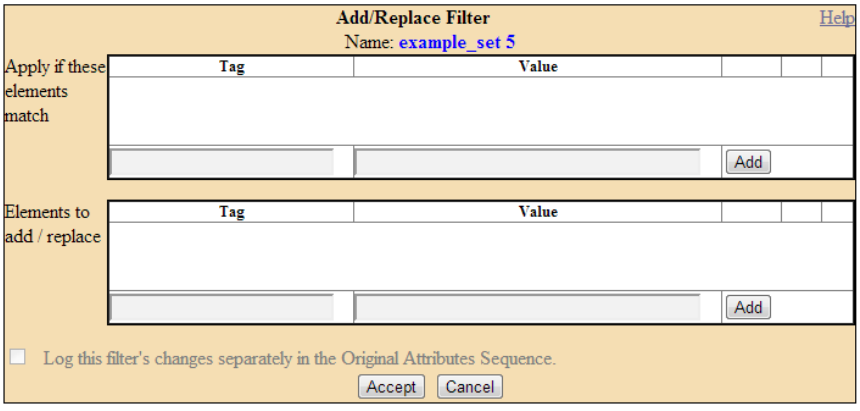
The DICOM Switchboard comes with a built-in class for editing DICOM Elements: DICOMElementFilter. This filter class is used to modify DICOM elements in a DIMSE message in various ways. For example, you can specify that a certain tag is always changed to another value, or that the data element contained within a particular DICOM tag is removed from the message. There are many ways to use the Element Filter, and not all are detailed here.

The Element Filter has five versions, but they are all subsets of the full-featured Element Filter. One (**Copy Filter**) allows you to copy elements in a DIMSE message, another (**Remove Filter**) will remove elements, and a third (**Add/Replace Filter**) will add or replace elements in a DICOM message (this is useful for replacing the value of an element); another (**Modify Filter**) allows you to modify the data in a DIMSE message with regular expressions and move and/or copy the new data to another field; the fifth (**Element Filter (full)**) allows you to do any or all of these things within a single rule. As the various types of element filters have subsets of the full filter's capabilities and are designed to simplify the editing process, only the most general cases will be described here.

When you add a new filter, you will see the list of the subsets of the Element Filter; choose one of the simplified versions, or choose the **full** version for the most power and flexibility – see **Figure 22** above. Examples of the screens you will see for each of the possible choices are shown below:







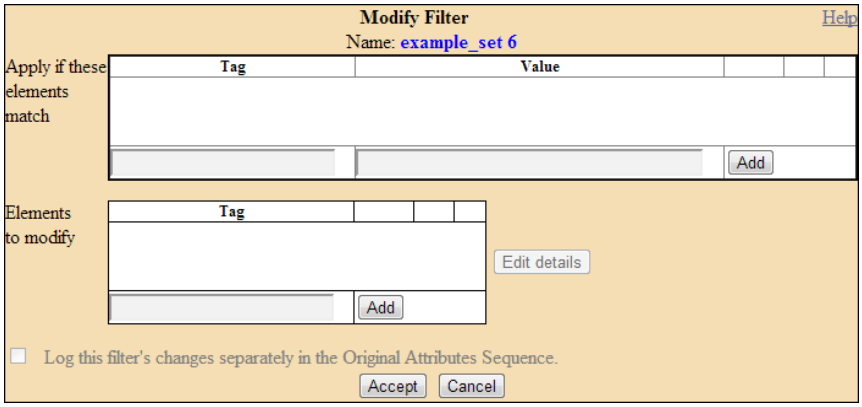


Figure 23 – Simplified versions of the Element Filters

Each of these Element Filters contains lists of elements that tell the filter how to process the data set. These elements are checked against the DICOM elements in a message and applied as appropriate. The rules are applied top-to-bottom, and the results of one rule are passed to the subsequent rules. In other words, the results of the first filter's actions are passed to the next filter, and so on in a serial fashion. For example, consider the case of a "copy" filter followed by a "remove" filter – the results of the elements that are copied will then be processed through the list of elements that are to be removed.

Note that the data that is displayed in these tables corresponds directly to the configuration groups in the actual filter configuration files that are usually located in the local file system in `$DCF_CFG\dicom\filter_sets`. ($DCF_CFG is the "cfg" directory under the Switchboard's installation directory, e.g., "`C:\LB Switchboard\cfg`" – see **Figure 6**.)

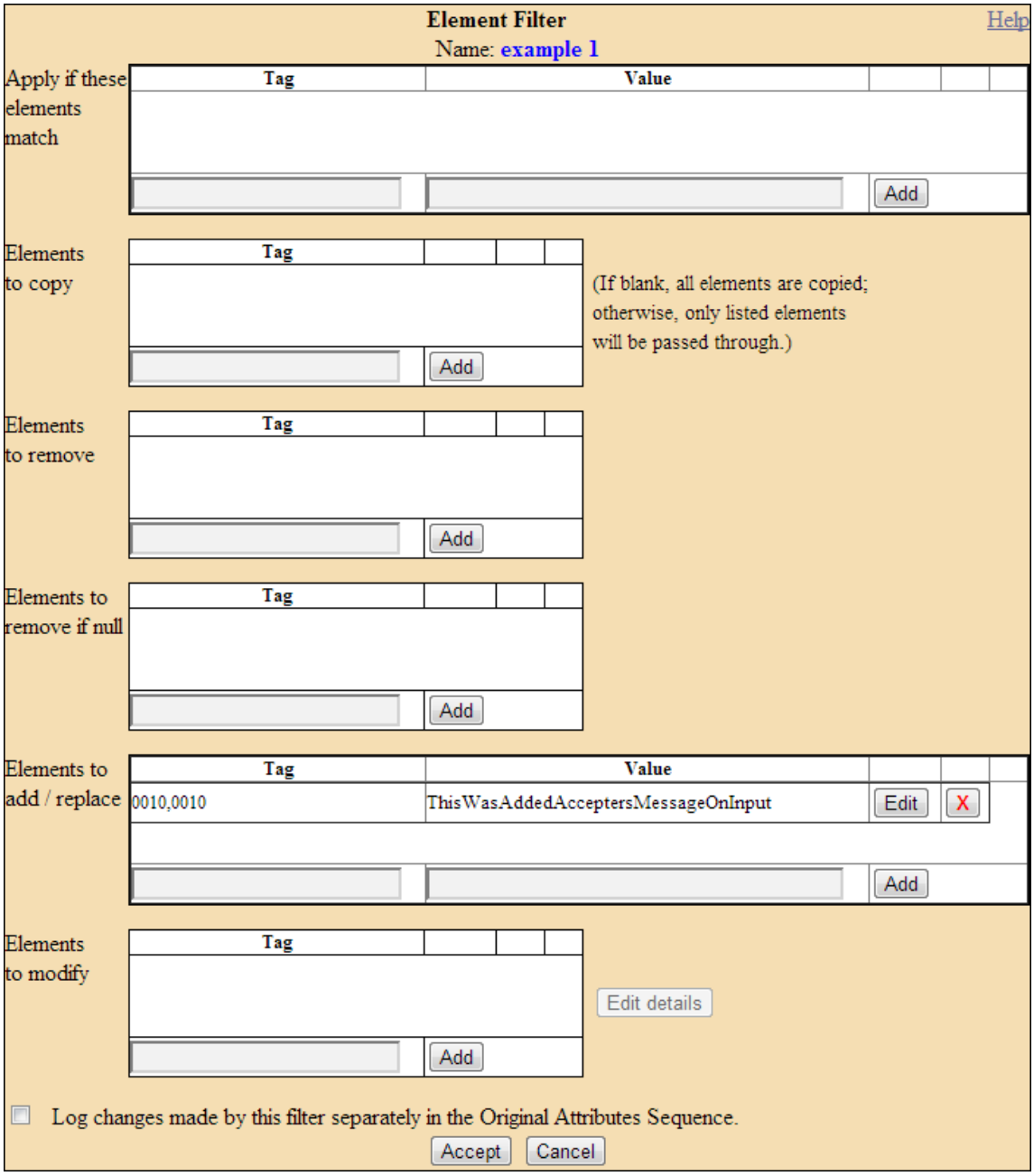


Figure 24 – Editing an Element Filter (Full version)

A filter may be selectively applied to only certain messages by specifying the **Elements to Match**; if Elements to Match is blank, the filter is always applied. If Elements to Match is present, then the message must include the tag with the given value, and the message will match, which will then cause the filter to be applied. (Note: all the simplified Element Filters also include this "Elements to Match" field, allowing them to also use this to determine when or if the filter should be applied.)

> **Note**: Elements to Match is usually used to make sure that a filter is applied if a certain *value* is *present* in a certain tag in the DICOM dataset to be filtered. But sometimes you may want to filter the data if a tag simply *exists* in the dataset, regardless of the value it has.
>
> Currently, for string VRs (such as PN, AE, CS, LO, SH, ST, LT, and UT) you can specify the value in Elements to Match as an asterisk ("*****") to check if the tag exists. (You could also

specify no value at all, but this is not the recommended syntax for checking if a tag exists.) Note that this only works for string VRs. To check if a tag exists for date/time VRs (DA, DT, TM), you would need to specify a range of dates or times that would match anything, e.g., `DA = 19000101-20200101`.

This will check if the tag exists and has any value. If you wish to check if the tag exists but has ***no*** value, this can be done with a few extra filters – see **Appendix A, section 5, Advanced Applications of the Modify Filter** for how to do this.

(Future releases of the Switchboard will include a more generic way to check if a tag exists, including a way that applies to binary and UI VRs.)

Elements to Copy indicates those tags that will be copied from the incoming DIMSE message to the outgoing DIMSE message; the elements' values will be unaltered. If the table is empty, all the elements in the message will be copied. **IMPORTANT**: If there are *any* elements listed in the table, *only* those elements listed will be copied into the new message.

Elements to Remove lists elements that should be removed from the incoming DIMSE message before it is sent on to the destination.

Elements to Remove if Null identifies elements that should be removed only if that element's value is present and is equal to *null*, i.e., empty. This filter allows you to remove empty elements from the dataset.

Elements to Add/Replace lists DICOM elements and their associated values that should be added to the message, or the new values to use for those elements, if they already exist in the data set. If an element listed in the table is found in an incoming DIMSE message, the value specified will be given to the element in the outgoing message, replacing the existing value. If an element listed is *not* found in the incoming message, then this element – with the specified value – will be added to the outgoing message.

The elements listed in the **Elements to Modify** table provide for the capability to modify the data in a tag in a DIMSE message. You specify the particulars of how the element data should be modified by selecting a tag in this table and then clicking the "*Edit details*" button, which becomes available after you enter a tag value. Selecting the "*Edit details*" button will change the screen to allow you to specify how the tag's data should be modified. Regular expressions may be used to match elements in a tag's value and rearrange the data. You may also move and/or copy the data to another DICOM tag, change the case of the data, and the like. See **Appendix A: Example Filter: Modifying Elements** for more information on modifying element data with filters.

Note on the Original Attributes Sequence: The DICOM Switchboard's filters, including the Element Filter, have the option to add data to the Original Attributes Sequence; this option is **on by default** at the **filter set** level. This can be used to create a "history" of the changes to the data resulting from the filtering by the DICOM Switchboard – the sequence will have the original values of the elements that were changed. When this is selected at the **individual filter** level, a separate entry is added to the OAS for each filter's changes. If this option is set at the **filter set** level, then just one entry is added to the OAS for all changes made by the filter set.

8.4.1 Mapping List Filter

The DICOM Switchboard's DICOM Mapping List Filter is a specialized and more advanced version of the Add/Replace Filter. It provides an efficient way to match a large number of possible values for a particular attribute (key tag), and then add/replace one or more elements, depending on the key tag value that is matched. This approach is useful and much more efficient if you have hundreds or even

thousands of possible substitutions – you can specify all substitutions in one Mapping List Filter instead of creating thousands of individual Element Filters that change a few tags only if the Elements to Match are matched.

The configuration for the Mapping List Filter is divided into two parts.

1. The standard DCF filter configuration that is read from the filter set config file. (This includes Elements to Match – see **Elements to Match** above for information on how this works.)
2. The Mapping List File: this is the full pathname for what is normally a plain text file with comma-separated values corresponding to the "key tag" and the "replace tag" elements. The name of this file, as well as the description of each column of data, is contained in the standard configuration group (item 1 above). You may also specify what character is used to separate the fields in the Mapping List File; any values apart from comma, semi-colon, and tab must be manually entered into the filter's configuration file.

When this filter is applied, the "key tag" is checked against the key tag values in the Mapping List File; if a match is found, the corresponding replace tag values are used to replace those tag values in the data set being processed.

For example, let's say that your "key tag" is 0008,0050 (Accession Number), your "replace tag" is 0020,000D (Study Instance UID) (note that you can have more than one), and the data in your Mapping List File is:

```
12345, 1.2.3.4.5
45678, 4.5.6.7.8
...
```

The first column in your Mapping List File corresponds to the key tag (0008,0050), and the second to the replace tag (0020,000D). (If you had multiple replace tags, you would have additional columns of data.)

If this filter is active and if the key tag in a dataset passing through the DICOM Switchboard has the value "12345", then the Study Instance UID would be replaced with "1.2.3.4.5"; if the key tag has the value "45678", then the Study Instance UID would be replaced with "4.5.6.7.8", and so on.

When configuring this filtering option there are two additional behaviors you may configure. First, you need to select the behavior when no matching "key tag" is found in the Mapping List File. There are three choices when no matching key tag is found:

- Reject the data set by aborting the association;
  the data set is not forwarded to the destination.
- Log a warning message and forward the filtered data set to the destination.
- Ignore the error and forward the filtered dataset to the destination.

Second, you need to select whether you will record these changes in the Original Attributes Sequence element of the data set being forwarded to the destination. If this check box is selected or if this option is set at the filter set level, then the changes made are added to the contents of the Original Attributes Sequence element.

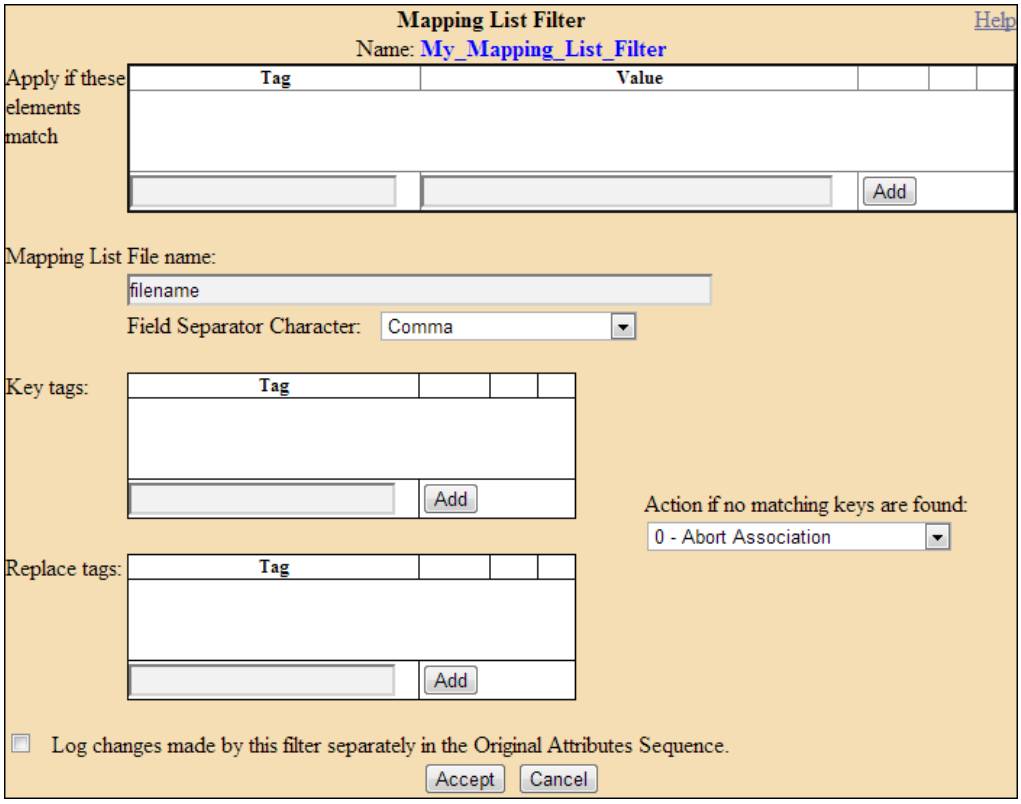


Figure 25 – Mapping List Filter

Note that you should enter the complete pathname for the Mapping List File, e.g., "`C:/home/MyStuff/data_mappings.txt`".

It is possible to have multiple match tags – in this case, the replacement occurs only if the values of each of the specified match tags match values in the mapping configuration file. If *all* of them match, then the replacement values are used to modify the dataset. If *any* of them do *not* match, then the replacement does not occur; in such a case, the resulting behavior is determined by the `no_match_option`. (Note that the matching values in the mapping list file will have any leading or trailing whitespace removed before the matching occurs. Do *not* depend on the values to match having either leading or trailing whitespace.)

For example, consider this filter configuration:

```
 [ mapping list ]
filter_type = DICOM_MAPPING_LIST_FILTER
match_tag = 0008,0050
match_tag = 0010,0010
replace_tag = 0020,000D
replace_tag = 0010,0010
mapping_cfg_name = /tmp/bob
no_match_option = 0
mapping_cfg_format = CSV
mapping_cfg_delimiter_char = ","
create_original_attributes_seq = FALSE
```

and its corresponding mapping list file

```
MY_ACCESSION_NUMBER_1, JOHN DOE, 1.2.3.4.5, John^Q^Public
MY_ACCESSION_NUMBER_2, JANE WOE, 1.2.3.4.6, Jane^Citizen
```


- Dataset #1 is to be filtered; it has 0008,0050 (accession number) with a value of "MY_ACCESSION_NUMBER_1", and its patient name (0010,0010) is "JOHN DOE". In this case, the Study Instance UID (0020,000D) will be set to "1.2.3.4.5" and the patient name will become "John^Q^Public".
- Dataset #2 is now to be filtered; its accession number is "MY_ACCESSION_NUMBER_1" and its patient name is "GARY DOE". Since one (or more) of its values do not match, the association will be rejected.
- But suppose dataset #3 has an accession number of "MY_ACCESSION_NUMBER_2" and a patient name of "JANE WOE". All values will find matches in the mapping configuration, and so the Study Instance UID will become "1.2.3.4.6" and the patient name will be changed to "Jane^Citizen".

Note that you can have as many match tags and as many replace tags as you require. In the mapping configuration file, the first **n** values on each line will be used for the mapping values, while the remaining **m** values will be the replace values. The order of the values on each line in the mapping file is important – the first value will be matched against the first specified match tag, the second value will be matched against the second match tag, and so on. Similarly, the values after the match tags will be the replacement tags – the first value will be used for the first replacement tag, the second value for the second replacement tag, and so on. Consider the above example, where the first two values are to be matched, and the remaining two are the replacement values.

8.4.2 Element Composer Filter

The Element Composer Filter is a more advanced version of the Modify Filter described above. It uses regular expressions to parse values from DICOM tags and combine the values into other DICOM tags. For example, you could take parts from two different tags and combine them to make a new value in a third tag. You specify regular expressions to parse each input tag and then how the patterns produced are combined to produce the output tags.

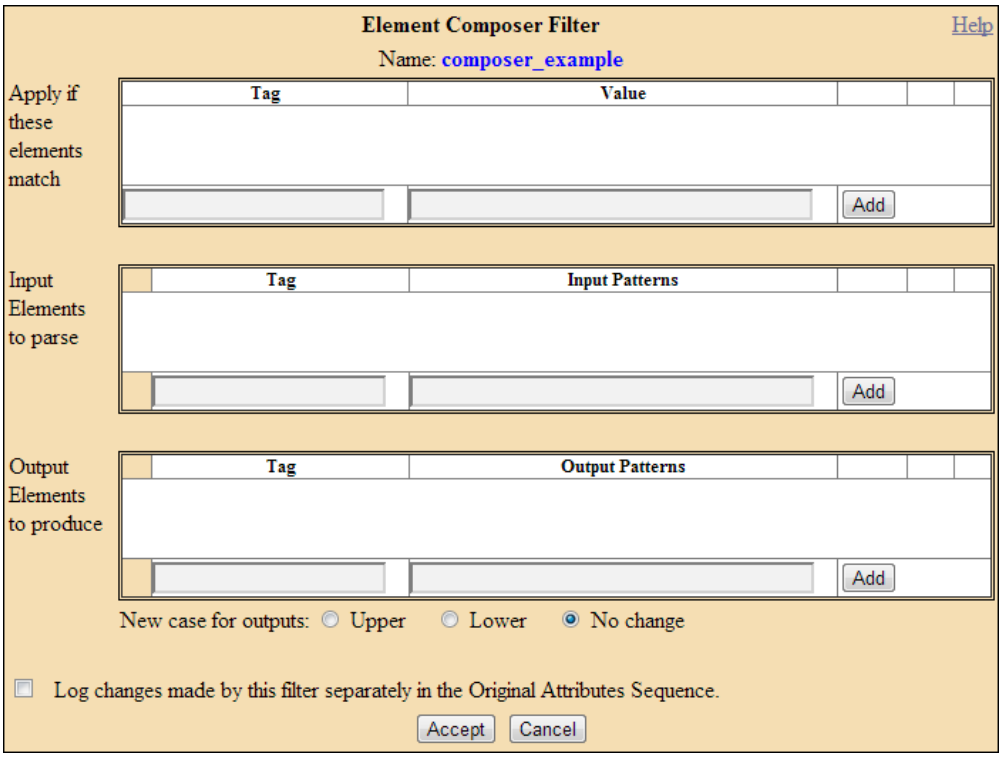


Figure 26 – Editing an Element Composer Filter

In the "Input Elements to parse" table, you 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 to produce" 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 Patterns specify what parts of the input elements to combine and how to combine them; note that you can use the parts multiple times and also combine them with plain text.

Appendix A: Example Filter: Modifying Elements has information on regular expressions and how to use them.

Note that the Element Composer Filter also includes Elements to Match – this determines if the filter should be applied, before any regular expressions are used. See **Elements to Match** above for information on how to use this.

8.5 Filtering Pixels

The DICOM Switchboard also includes a **Pixel Value Shift** filter that can be used to shift the bits in pixel data values left or right, for image data manipulation. (This filter is primarily used in unit/integration tests for data-set and DIMSE-message filtering, but it is also available for real world image data manipulation.) The **Shift bits** value indicates how many bits to left-shift each byte or word (16 bit) sample in OB or OW pixel data values; negative values indicate a right shift.

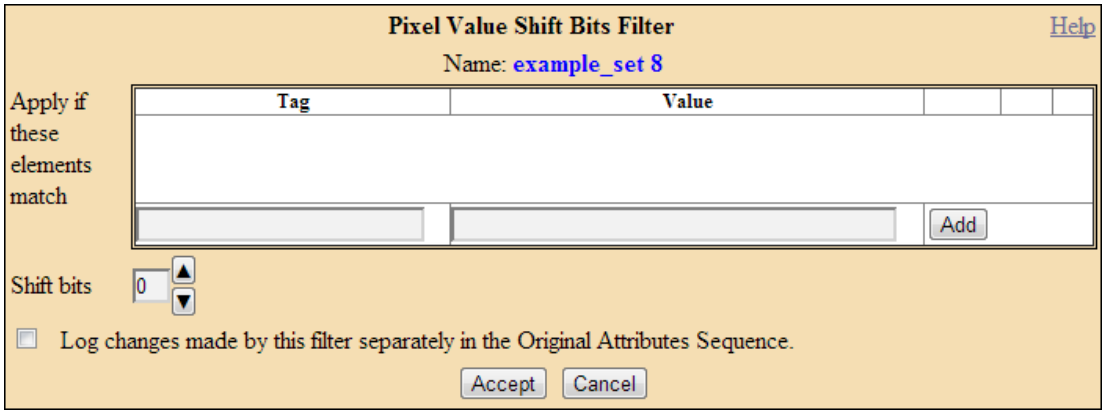


Figure 27 – Editing a Pixel Value Shift Filter

Please note that if you are using a Pixel Value Shift Bits Filter, you will need to change the Switchboard's configuration to have the image data be decoded – by default, the image data is passed through as-is and is not decoded.

To reconfigure the DICOM Switchboard to allow a pixel filter to be applied, use a text editor (such as Notepad or VIM) to edit the file "`$DCF_CFG/apps/defaults/dcf_switch`" ($DCF_CFG is the "cfg" directory under the Switchboard's installation directory, e.g., "`C:\LB Switchboard\cfg`" – see **Figure 6**). Within that file, find the section "`java_lib/DCS/default_session_cfg`" and within that section, change the value of the attribute "`enable_compression_pass_through_mode`" to "`no`"; also change the value of the

attribute "`enable_streaming_mode`" to "`no`". Save the changes to the file and restart the DICOM Switchboard.

Note that the Pixel Value Shift Filter also includes Elements to Match – this determines if the filter should be applied. See **Elements to Match** above for information on how to use this.

8.5.1 Converting Planar Configuration

The **Planar Configuration Convert** filter can be used to convert a color image from interleaved format (RGB RGB RGB…) to planar (RRR…GGG…BBB…), or vice versa. Note that this filter will not modify the data unless samples-per-pixel is greater than 1, and bits-allocated is 8. This filter will recognize pixel data that is stored in attribute 7FE0,0010 in the top-level data set, as well as pixel data that is contained in either of the attributes Basic-Grayscale-image-sequence or Basic-Color-image-sequence.

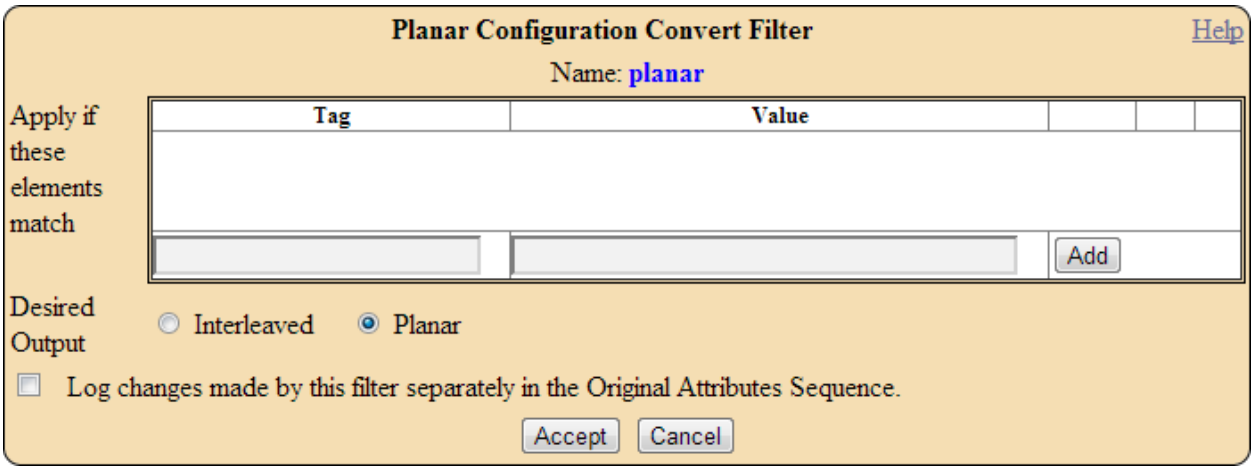


Figure 28 – Editing a Planar Config Convert Filter

8.5.2 Fix Icon Image Sequence

Some images may have a compressed Icon Image Sequence even though the rest of the data is uncompressed – this is invalid. If you are sending data that has a bad Icon Image Sequence, the **Fix Icon Image Sequence** filter can try to fix the bad sequence – it will try to read the data with various transfer syntaxes, and then it will write the icon image as uncompressed data. If it can't decode the icon, it will delete the bad sequence. (This filter does not have any configurable attributes.)

8.6 Padding Values

The DICOM Switchboard's Pad Value Filter can be used to pad a string value with a character until the string is a given length. For example, you can make sure that the Accession Number is a certain length and has leading zeroes, or make sure that the Patient Name is padded with spaces until it is a certain length.

Figure 29 – Editing a Pad Value Shift Filter

The **Tag to Pad** is the tag whose value should be padded – its value will be retrieved, checked if it is long enough, and padded until the appropriate length is reached. **Desired Length** is how long the string should be after it is padded. Please note that this is *not* how many pad characters should be added, nor is it the maximum length of the values – any values longer than this will not be padded at all. You can choose if the string should be padded with *leading* pad characters by choosing the **Left** button, while the string will be padded with *trailing* characters by clicking **Right**. The Pad Character can be a space or a null character by clicking the appropriate (**Space** or **Null**) box; if you want to pad with a different character, click the **Other** button and enter the character in the box to the right.

For example, suppose you want the Patient Name (tag 0010,0010) always to be at least 20 characters long and have leading zeroes as part of it. You would enter "0010,0010" in the Tag to Pad field, "20" in the Desired Length field, and click the Left button; then you would click Other and enter "0" in the box. This would result in the value "`John^Doe`" becoming "`00000000000John^Doe`".

Or suppose you want trailing characters instead of leading zeroes – you get this by clicking the Right button. Then "`John^Doe`" would become "`John^Doe00000000000`", while "`John^Philip^Sousa`" would become "`John^Philip^Sousa000`", and "`John^Jacob^Jingleheimer^Schmidt`" would be unchanged, since its length is more than 20 characters.

Note: You should be careful if you are padding a value with nulls, spaces, or other whitespace characters. Such characters can be stripped off when the value is transmitted or written, since many VRs regard whitespace as insignificant.

Also, please note that private tags (VR: UN) should be added to the Extended Data Dictionary so that their VR can be found and so that values are correctly processed as strings. By default, an element with VR: UN is interpreted as binary data, which should not be processed as a string.

Note that the Pad Value Filter also includes Elements to Match – this determines if the filter should be applied. See **Elements to Match** above for information on how to use this.

8.7 Entering Data

Data is entered in the "Tag" fields as a pair of hexadecimal (base-16) numbers, separated by a comma, e.g., `0020,0020`.

Once a DICOM tag has been entered in the left-hand "Tag" column, the values should be entered into the right-hand "Value" column. Then click the "Add" button to add the tag and/or value to the table. (See **Appendix A: Section 4.1, Entering Sequences** for information on entering a sequence as the tag.)

To edit a tag or a value in a table, click the "Edit" button next to the tag or tag-value pair in the table. You will be prompted for the new value(s).

To delete a tag (or tag-value pair) from a list, press the small red "X" next to it in the table.

After you are done editing the filter's data, click the "*Accept*" button to accept the changes and save them. Press the "*Cancel*" button if you wish to discard the changes you have made.

Note: after you have become accustomed to editing filters with the DICOM Switchboard's Filter Editor, you may wish to look at your filter's actual saved configuration data in a text editor for comparison purposes. As you learn how to create and edit filters, there may be some cases where you may find it easier to use the text editor to modify the configuration data directly, rather than using the GUI-based Filter Editor. The actual filter definitions are stored in configuration groups within the filter configuration files that are usually located in the local file system in `$DCF_CFG\dicom\filter_sets`. ($DCF_CFG is the "cfg" directory under the DICOM Switchboard's installation directory, e.g., "`C:\LB Switchboard\cfg`" – see **Figure 6**.)

8.7.1 Using Macros to Specify Data

As you use the DICOM Switchboard to filter data, you may encounter situations where the new values that you want are dynamically changing. For example, you may wish to specify that a tag should have the current date and time. Obviously, you can't specify the current date and time exactly as a text string, since the date and time keep changing. Instead, the DICOM Switchboard provides *macros* to fill in values that are changing.

The DICOM Switchboard provides the following macros for your use:

- ${DATE} – the current date in the format YYYYMMDD, e.g., "20071108" (November 8, 2007)
- ${TIME} – the current time in the format HHmmSS, e.g., "142035" (2:20:35 PM)
- ${DATETIME} – the current date and time in the format YYYYMMDDHHmmSS, e.g., "20071108142035"
- ${GMT_TIME} – the current time for Greenwich Mean Time in the format HHmmSS, e.g., "182035" (6:20:35 PM GMT/UTC)
- ${GMT_DATETIME} – the current date and time for Greenwich Mean Time in the format YYYYMMDDHHmmSS, e.g., "20071108182035"
- ${TZ} – the current time zone, e.g., "Eastern Daylight Time". Note that the full name, not an abbreviation is returned. (For Linux Switchboards, "EDT" is returned, not the full name.)
- ${TZOFFSET} – the offset of the current time zone from GMT, e.g., "-0400" for EDT
- ${UID} – generates a new **U**nique **ID**entifier

Note that the times specified are in local time unless you use the GMT macros.

You can use these macros to specify the new values for tags just the same way as if you were specifying the exact text. For example, to change the E_INSTANCE_CREATION_DATE (0008,0012), you would specify the new value as "${DATE}". You can also have a value that mixes text and macros. Let us suppose you wanted to change the username to be "Bob <current time>"; you would set the new value to be "Bob ${TIME}", which would give you a result something like "Bob 150721".

If you wish to specify a value that has a dollar sign ("$") in it and not have it interpreted as a macro, you should escape it with a backslash, e.g., "\${UID}"; this would insert the string ${UID} literally in the value.

These macros greatly increase the flexibility and power of the DICOM Switchboard's filters and allow you much more capabilities in how your data is filtered. For example, if you wanted to specify a sequence that has known information but also dynamic information like the date, you could create a filter to insert the literal values but also uses the macros to set the date and time.

8.8 Example Filters

- Example 1: Replacing a Value
- Example 2: Removing an Element
- Example 3: Modifying an Element's Value with Regular Expressions
- Example 4: Composer Filter Examples

8.8.1 Example 1: Replacing a Value

Suppose you have a DICOM data set that has an incorrect value, and you want to use the DICOM Switchboard to correct that value. For example, if the patient name was "Public^Jane^Q" but was supposed to "Doe^John", then you could set up a filter to replace the incorrect value every time it passed through the DICOM Switchboard.

Via the Filter Editor, you would choose to create a new filter, choosing the filter type to be "Add/Replace Filter". If you only want to correct the element when it has the value "Public^Jane^Q", you would first enter into the table for "elements to match" the following values:

Tag	Value
0010,0010	**Public^Jane^Q**

If you want to correct the element every time, even when the value is not "Public^Jane^Q" or the element is not present at all, you would leave this table empty.

Secondly, you would enter into the table for "Elements to add/replace":

Tag	Value
0010,0010	**Doe^John**

8.8.2 Example 2: Removing an Element

If you want to remove a DICOM element from a data set, you would create a new filter with the type "Remove Filter". For example, let us suppose that you always want to remove element 0028,0010 (rows). You would enter into the table for "elements to remove" the following values:

Tag	**0028,0010**

You could configure the filter to be applied under certain conditions – i.e., when an element has a certain value – by filling in the "Elements to Match" table (as in the previous example); or you would leave "Elements to Match" blank to apply the filter all the time.

8.8.3 Example 3: Modifying an Element's Value with Regular Expressions

The "Modify Filter" is designed to allow you to modify the value of an element by using a regular expression (or a "regex", as it is commonly called); the "Modify Filter" also allows you the option to move or copy one element's data into other elements.

For example, suppose that your receiving software expects the patient's name to be in your proprietary DICOM tag "`abcd,abcd`", but you are not getting that tag sent from the modalities. In addition, suppose that the data has the first name first, instead of the DICOM default of last name first. You could use the Modify Filter and regular expressions to switch the elements around and move it to the desired tag.

First, you would enter into the table for "Elements to modify" the tag value for the standard DICOM Patient Name tag:

Tag	**0010,0010**

Next, you would select the tag that you just entered and click the "*Edit details*" button; this will open up a new window, where you may indicate how the data for that tag should be modified. (See **Appendix A: Example Filter: Modifying Elements** for more information about creating Modify Filters.)

In the new window, you would specify how the regular expression should modify the data. (In this example, we don't describe how to write a regex for this purpose. Details are covered later in **Appendix A: Section 3, Modifying Data Using Regular Expressions**.) In this case, you would enter the following:

Old value	**([^]*)\^([^]*)**
New value	**$2\^$1**

Using the built-in test feature you can even test your regex by supplying sample strings to see that "john^doe" becomes "doe^john", for example.

Second, to copy the modified result to your private tag as proposed in this example, you would click the checkbox for "*Also Copy data to tag*" and enter your proprietary tag – "`abcd,abcd`" – in the box. The end result is that the order of the first name and last name will be swapped, and the data will be copied to your tag.

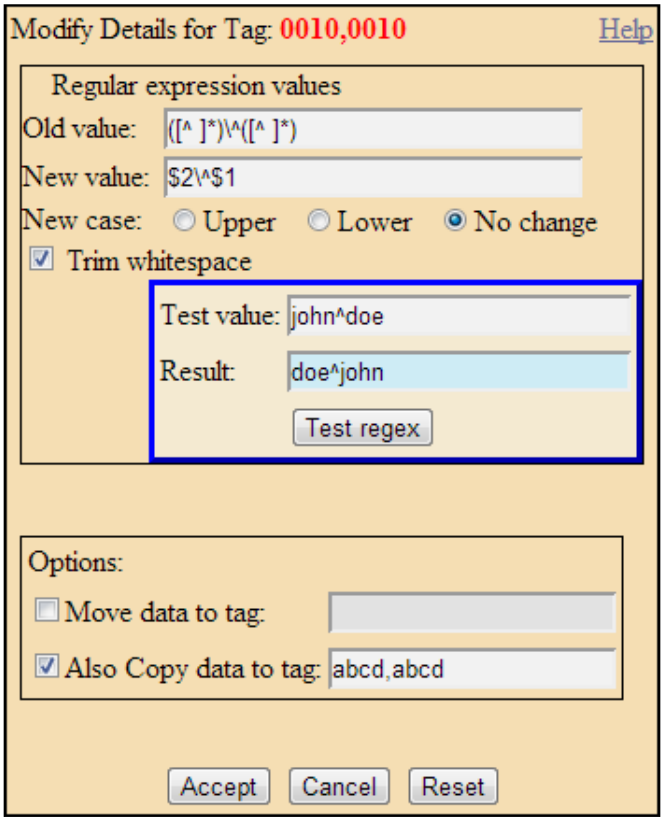


Figure 30 – Details of a Modify Filter

8.8.4 Example 4: Composer Filter Examples

The Element Composer Filter uses regular expressions to parse an element's value and combine the parts into new elements. When you create a new filter, you would select the "Element Composer Filter" as the type and then enter the following data, depending on what you want to combine.

1) **Swap two tags** – Let's say that you want the patient's name and the patient's ID to be in each other's place (yes, I know this is a bad example – work with me…). You would specify the following inputs:

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

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

Tag	Output Pattern
0010,0010	${2.1}
0010,0020	${1.1}

This results in the first pattern ${1.**1**} from the first input tag ${**1**.1} being put into the second output tag (0010,0020), and the first pattern ${2.**1**} from the second input tag

${2.1} being put into the first output tag (0010,0010). (In this case, the first pattern is also the entire value.) So if you started with "John Doe" and "1.2.3.4.5" in Name and ID respectively, your result would be a Patient ID of "John Doe" and a Patient's Name of "1.2.3.4.5".

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

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

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

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

This means that the first pattern – the first 10 characters – will go into the Accession Number; everything else from the Accession Number will go into the Requested Procedure ID. So if your 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** – Suppose you want to take parts of the Accession Number and parts of the Requested Procedure ID and "mix and match" them.

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

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

Then the outputs might look like this:

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

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

9 Switchboard Mapping Rules

The Mapping Rules used by the DICOM Switchboard tell it where and how to route the messages it receives.

From the Switchboard Operations Window, clicking the "*Edit Mapping Rules*" link (on the Configuration tab) will start the Switchboard's Mapping Rule Editor web app.

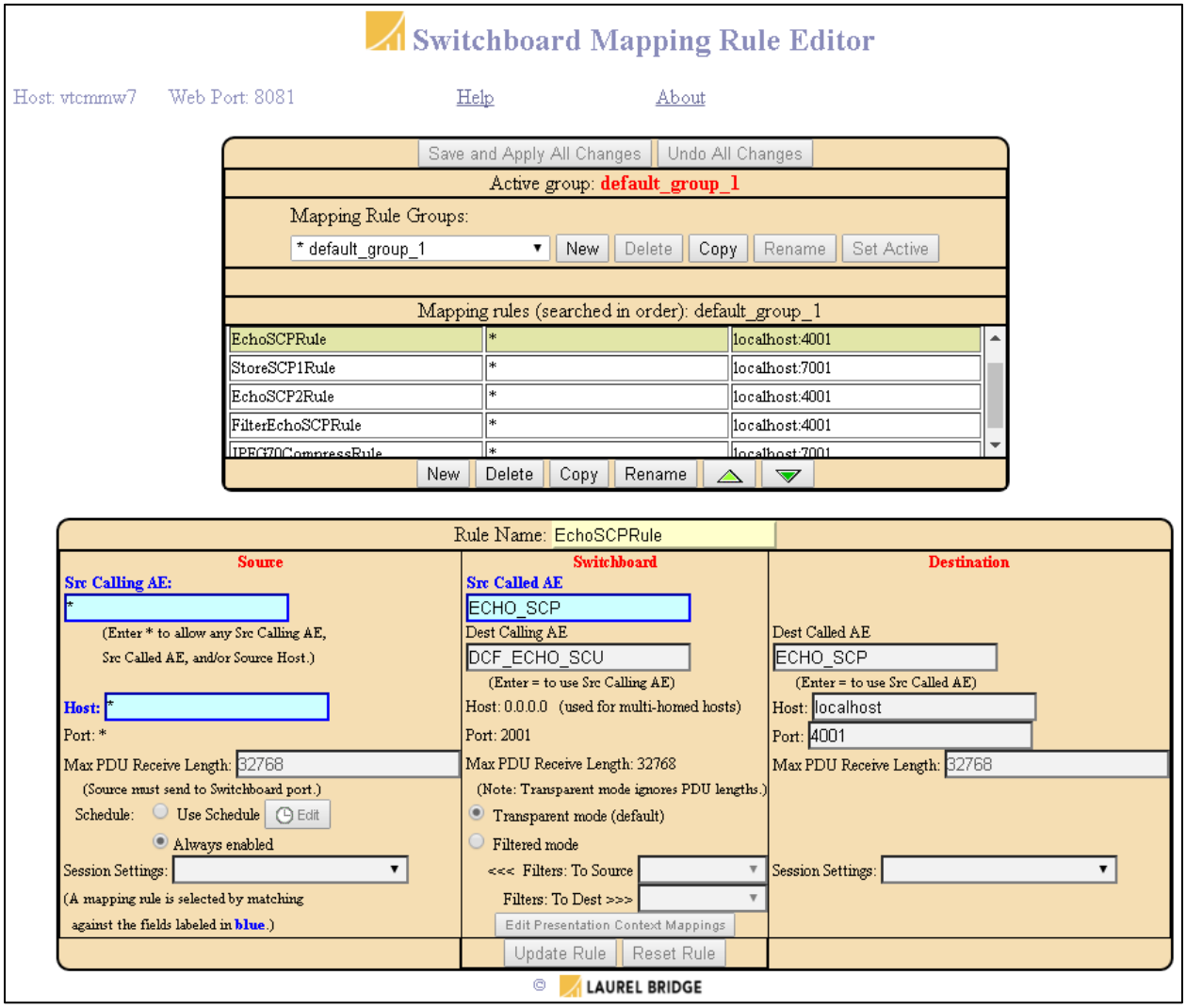


Figure 31 – Mapping Rules Editor Screenshot

Note: As of version 3.3.40c, the Switchboard uses a non-applet version of the Mapping Rule Editor that is more browser-friendly. However, the old Java version is still available and may be used in a non-applet mode.

For Linux: The Mapping Rule Editor (Java app) may also be launched from a Switchboard Linux Command Terminal by running the command "`bin/Run_MappingRuleEditor.bash`".

Note: On Windows the Mapping Rule Editor (Java app) may be run via the command "`Run_MappingRuleEditor.bat`" from a Switchboard Command Prompt.

You will have to add new mapping rules or edit the existing mapping rules as necessary for your systems. You will have to modify the rules in order to indicate what AEs (Application Entities) are handled by the DICOM Switchboard and where the data for such AEs is sent. To incorporate filtering into a source device's communication with a destination device, you will need to configure the source to send messages to the DICOM Switchboard instead of to the original destination host, and you will have to configure the DICOM Switchboard to forward those messages to the appropriate destination; you will also need to attach the desired filters to the mapping rules.

Within the DICOM Switchboard, DICOM messages are matched according to several criteria and forwarded appropriately. For example, the DICOM Switchboard can be configured so that a message requesting a Called AE of EchoSCP is sent to destination host "alpha" listening on TCP port 123, while a message requesting a Called AE of StoreSCP2 is sent to destination host "beta" listening on TCP port 456.

As another example, you might set up the Switchboard's mapping rules so that all messages received with a Called AE Title of "StoreSCP4" are sent to the destination's host and port with that same Called AE.

You can also specify that some rules should be applied only on certain days and times – for example, you could say that a certain AE is sent to one destination during business hours and then create another mapping rule to send that AE to a different destination the remainder of the time. (**Note**: this capability should be used with caution since it is easy to create a configuration with schedules that do not cover every time period. In such a case, a connection might be rejected by the Switchboard even though you *think* it should be accepted.) This capability is described in Section **9.2.6** below.

The Switchboard's Mapping Rule Editor simplifies the editing of these mapping rules and allows you to optionally choose the filters that should be applied to the messages that match a particular mapping rule. You may create your own mapping rules by selecting the "*New*" button below the list of Mapping Rules. When you select this option, you will be prompted to the name of the new rule; a new rule will be created with default values, and the new rule will be displayed in the table at the bottom for editing:

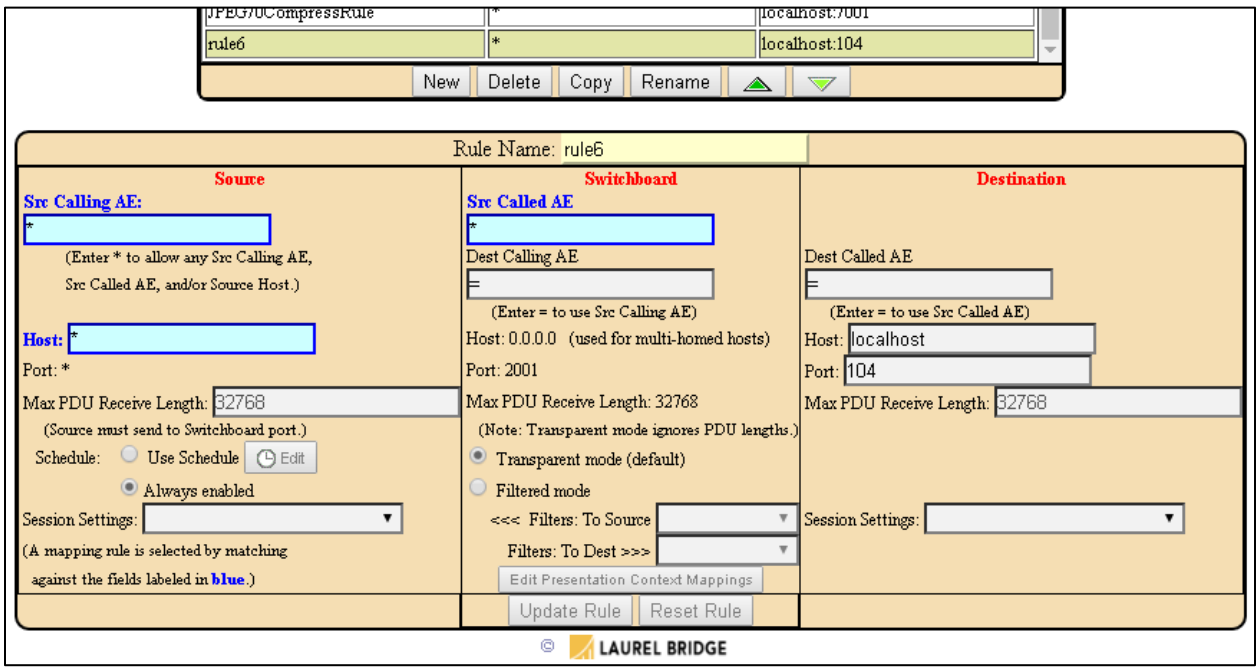


Figure 32 – Add a new Mapping Rule

If any of your mapping rules use "Filtered Mode", you may select the global filter sets to use via the drop-down lists in the Switchboard column. You should click "*Edit Global Filter Sets*" from the Switchboard Operations Window to edit the global (shared) filters. Filters must be associated with a Mapping Rule in order for data passing through the Switchboard to be altered.

9.1 About Mapping Rules

The DICOM Switchboard interposes itself between SCUs and SCPs, allowing the messages between the clients (SCUs) and servers (SCPs) to be monitored, viewed, and filtered. To do this, the DICOM Switchboard masquerades or pretends to be the SCP for the clients, and then masquerades or pretends to be the SCU for the servers. The Switchboard's mapping rules tell the DICOM Switchboard where and how to route the messages it receives.

As an example, suppose you have a client (SCU) that is trying to store data to a destination server (SCP), the DICOM Switchboard acts as the SCP for the client SCU, while the DICOM Switchboard acts as the SCU to the *actual* destination server (SCP). To accomplish this you might have the following configuration settings:

Source Calling AE	*	Destination Calling AE	StoreSCU
Source Called AE	StoreSCP1	Destination Called AE	StoreSCP1
Source host	*	Destination host address	alpha:1234

This configuration means that the DICOM Switchboard will listen to any client (Source Host = *) with any Calling AE (Source Calling AE = *) that wants a Called AE of StoreSCP1 (Source Called AE = StoreSCP1). The client (SCU) sends to the DICOM Switchboard, the DICOM Switchboard filters the data and forwards the message to the real destination server (SCP) by acting as the SCU (Destination Address = alpha:1234). The forwarded message will have the Calling AE and Called

AE as specified by the other Destination values. Any messages from the real SCP to the real SCU will be forwarded back through the DICOM Switchboard as long as the association – which is maintained by the Switchboard – remains active. This configuration translates to a mapping rule with parameters set like that shown in the table below.

Source	DICOM Switchboard	Destination
Src Calling AE: *	Src Called AE: StoreSCP1	
	Dest Calling AE: StoreSCU	Dest Called AE: StoreSCP1
Host: *		Host: alpha
		Port: 1234

The table above shows, in a general way, the path of the data in a connection through the DICOM Switchboard. The source is on the left, the DICOM Switchboard is in the middle, and the destination is on the right.

- Data comes from any source host into the DICOM Switchboard, with the source host calling for an SCP named StoreSCP1.
- Data goes through the DICOM Switchboard.
- The DICOM Switchboard sends the data on to the destination host alpha at port 1234, using a destination Calling AE of StoreSCU and a destination Called AE of StoreSCP1.

When you enter this data into the Mapping Rule Editor, it should look like this screenshot:

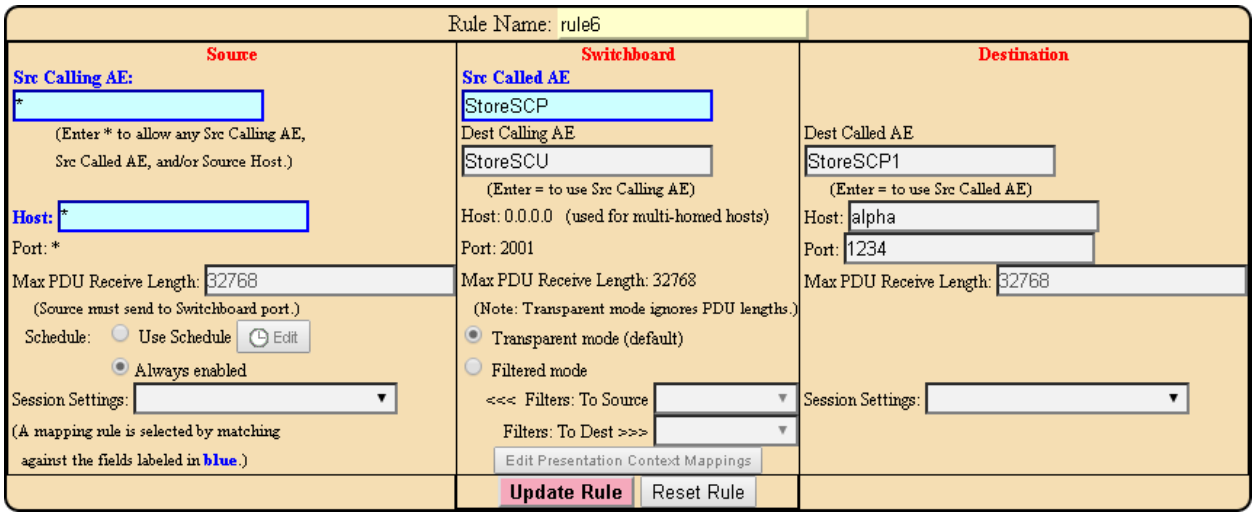


Figure 33 – Example Mapping Rule

If you need many similar mapping rules (e.g., hundreds or thousands), see **Appendix H: Using & Creating Many Mapping Rules** for information on how to use scripts to create them in batch mode.

9.2 Mapping Rule Editor

The Switchboard's Mapping Rule Editor is used to configure the mapping rules that are used by the DICOM Switchboard. The mapping rules are organized into groups. This approach allows a different

set of rules to be selected based on your setup and circumstances. For example, you can have one group of rules to be used at one site, and another group is available for another site – you just select which group you want to use depending on where you are running the application. As another example, you could have one set of rules for use in your testing, and another set of rules for actual operations; then, if you have a problem with the actual operations, you can quickly and easily switch to the testing group of rules to try to diagnose the problem.

9.2.1 Activating New Mapping Rules

As you create and edit the mapping rules, the changes you make are stored locally in the Mapping Rule Editor until you click the "Save and Apply All Changes" button located at the top of the window. This will commit the changes you have made to the application's configuration database, and it will also load the new mapping rules into the Switchboard's internal routing system.

Note: Any changes you make to the mapping rules or mapping rule groups will *not* take effect until you click the "Save and Apply All Changes" button. You *must* commit your changes with this button in order for your changes to be saved to the configuration database and used by the DICOM Switchboard. This applies to any changes you make: adding, deleting, or editing rules or groups.

Click the "Undo All Changes" button to undo any changes you have made to the mapping rules or mapping rule groups, since the last time that they were committed. Once the changes are committed, you can no longer reverse them by clicking "Undo All Changes".

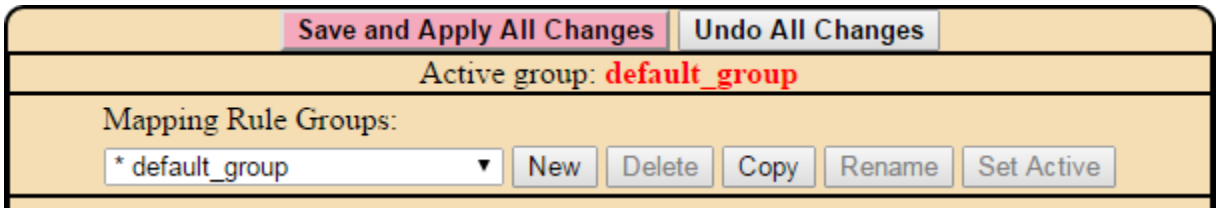


Figure 34 – Committing your Changes

9.2.2 Rule Group Options

Rule Groups are listed and selected in the drop-down list in the top panel of the Mapping Rule Editor.

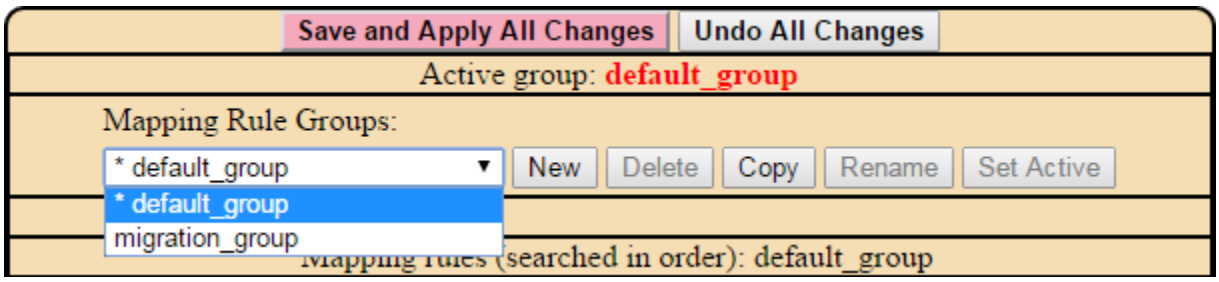


Figure 35 – Mapping Rule groups

The Active group (that is, the set of rules that will be used by the DICOM Switchboard) is shown at the top of the display and is marked with a "star" in the list – in the illustration above; it is called "**default_group**". Below that is a list of other groups of rules that are available for use. Selecting a group will cause its rules to be displayed in the table below the list. You may add a new group,

delete, copy, or rename a group, or change the active group by using the buttons to the right of the list. You may need to create a new group of rules and add rules to it or edit the existing rules in the groups to get the desired mapping configuration.

The functions of the various option buttons are described below:

- **Select the group of mapping rules to use** – If the currently active group is not the one you wish to use, select the desired group and press the "*Set Active*" button. A "star" beside the name in the list indicates the active group.
- **Adding a group** – You may create an empty group by clicking the "*New*" button and entering the name for the new group; then you must add rules to the empty group in order to use it. You also may copy the currently selected group by pressing the "*Copy*" button.
  Please note that the name must be unique – it cannot be the same as another group – and that certain characters, such as quotes and slashes, are not allowed.
- **Removing a group** – If you wish to remove a group of mapping rules, select the group in the list and click the "*Delete*" button. You will be asked to confirm the deletion of the group; press "*OK*" to continue deleting the group, otherwise press "*Cancel*".
  Please note that you cannot delete the active group.
- **Renaming a group** – Select the group that you wish to rename and click the "*Rename*" button; you will be prompted to enter a new name for the group.
  Please note that you cannot rename the active group, and the new name must be unique and not have certain special characters.
- **Reloading the Data** – The easiest way to reload the data used by the editor is to close the window and open it again by clicking "Edit Mapping Rules". You may also do it in most browsers by pressing the F5 key.
  Please note that reloading the data only refreshes the data being used by the Mapping Rule Editor, *not* by the underlying Switchboard itself; the DICOM Switchboard may need to be restarted in order for any changes to the rules to take effect.

Remember: Any changes you make to the mapping rule groups will *not* take effect until you click the "*Save and Apply All Changes*" button.

See Section **12.9** for instructions on how to import or export your groups of mapping rules.

Once you have selected a group of mapping rules, you may need to add mapping rules to it and/or edit the rules in it to achieve the mapping behavior that you desire. When you select a group of

mapping rules in the top panel, the rules in that group are displayed in the table below the list, as illustrated **below**.

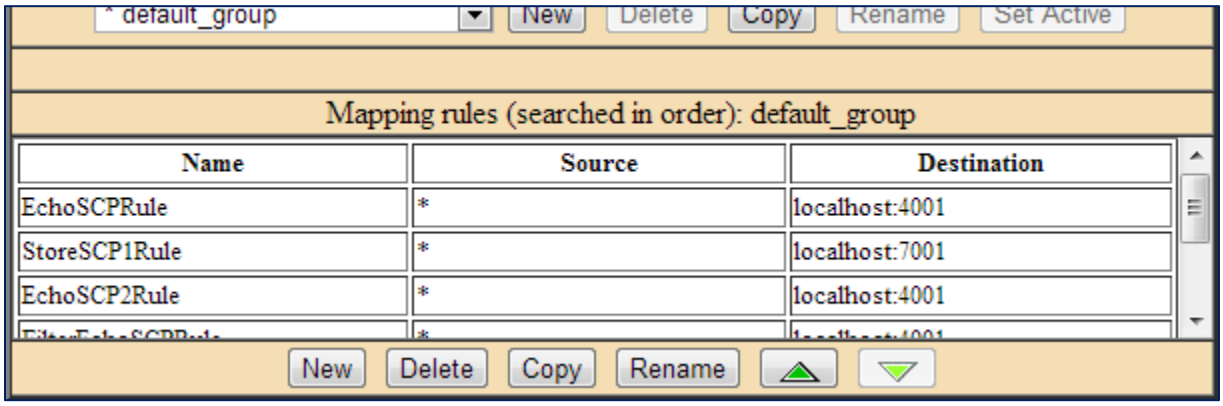


Figure 36 – Mapping rules in the group called "*default_group*"

9.2.3 Mapping Rules – Options

- **Editing a rule** – Selecting a rule from the list of rules in the table will display that rule's information in the bottom panel, as illustrated **below**.

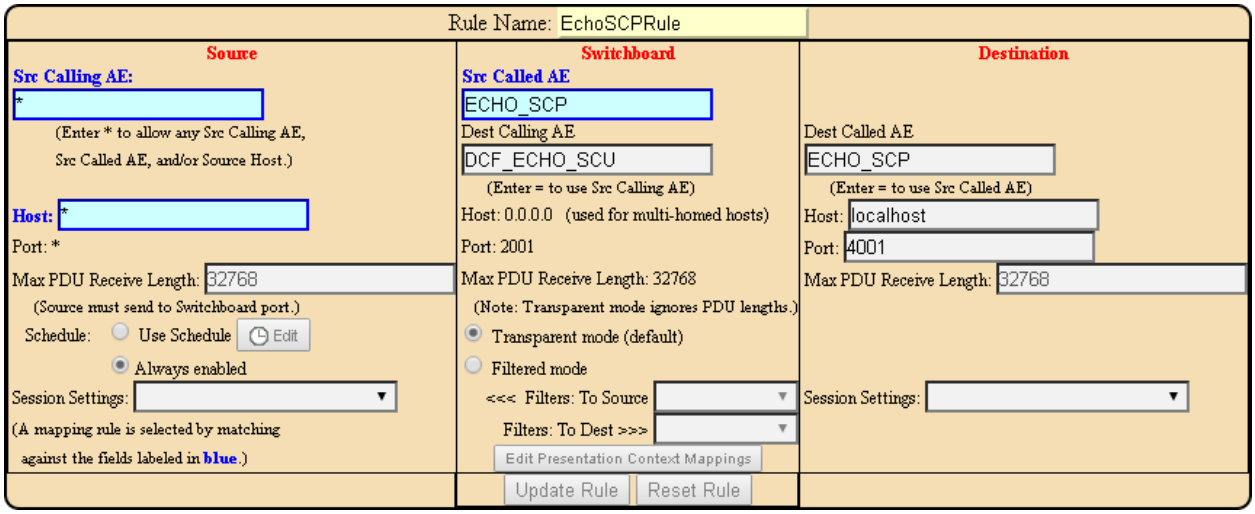


Figure 37 – The Echo rule's data is displayed for editing.

If you wish to edit that rule's information, you may change the data directly in the displayed panel. If you make a mistake, you may reset the data by pressing the "*Reset Rule*" button to restore the rule to its original values. Once the changes to that rule are completed to your satisfaction, store the changes by pressing the "*Update Rule*" button.

Note that you cannot change the name of a rule while editing the rule – this is done via the "*Rename*" button below the table of rules.

- **Adding a rule** – If you wish to add a new mapping rule, press the "*New*" button. You will be prompted to enter the name for the new mapping rule, and then the rule will be selected for editing (see **Figure 32** above). Please remember, however, that each rule *must* have a unique name, and certain characters (such as slashes) are not allowed.

- In the **Src Calling AE** box, enter the AE-Title of the calling system (source SCU), which is one of the match criteria for this rule; enter an asterisk (*) as a wildcard to match all calling AE-Titles.
- The **Src Called AE** box should be filled in with the name of the AE that the client is contacting, which is one of the match criteria for this rule; enter an asterisk (*****) as a wildcard to match all called AE-Titles.
- In the source **Host** box, enter the IP address or hostname of the source for messages that this rule should handle. If the messages can be from any source, enter an asterisk (*****) as a wildcard to match all. Note: source Host will be one of the match criteria for this rule.
- In the **Dest Calling AE** box, enter the AE-Title for the DICOM Switchboard to use when it forwards the message; entering "**=**" (equals sign) causes this value to be made the same as the Src Calling AE.
- The **Dest Called AE** box should be filled in with the AE-Title of the destination server; you may also enter "=" if this value should be made to be the same as the Src Called AE.
- In the destination **Host** box, enter the address where the messages should be sent – this can be either a machine hostname or a numeric IP address.
- In the destination **Port** box, enter the listening TCP port of the destination SCP.
- Uncheck the "**Always enabled**" checkbox next to "**Schedule**" *only* if you wish to specify when this rule should be applicable, and then click the **Edit** button to set its schedule. By default, mapping rules are always enabled and do not require a schedule.
- The **Session Settings** for Source and/or Destination specify files of advanced options that let you alter how the data is handled as it comes into Switchboard or goes out – for example, you can change the quality of the JPEG compression that is used. You select a file of session settings from the drop-down list. The session settings files must be created *manually* in the "`dicom/session_settings`" directory under the Switchboard's configuration ("`cfg`") directory. See **Appendix K: Session Settings** for the possible Session Settings attributes and their values.
- In the Switchboard column, selected Transparent or Filtered mode. If the rule uses Filter mode, select the global filter sets to use via the select lists. (See Section **9.2.8, Operational Modes: Transparent vs. Filtering** below for more information on *Transparent* vs. *Filter* mode and how to edit the filters.)

For example, if you have a client SCU that stores data through the DICOM Switchboard (SCP), while the DICOM Switchboard then acts as the SCU to the actual destination SCP, then you might have the following configuration information entered in your mapping rule:

Source host:	*****	**Destination host address**	Alpha
		Destination port	1234
Source Calling AE	*****	**Destination Calling AE**	StoreSCU
Source Called AE	StoreSCP1	**Destination Called AE**	StoreSCP1

This means that the client has called StoreSCP1 on the DICOM Switchboard, thinking that the DICOM Switchboard is the SCP it wants. The DICOM Switchboard forwards the message to the real SCP, **alpha**, with the DICOM Switchboard acting as the SCU with calling AE-Title of StoreSCU. This translates to a Mapping Rule entry that looks something like this when filled into the editor window:

Source	**DICOM Switchboard**	**Destination**
Src Calling AE: *****	Src Called AE: StoreSCP1	
	Dest Calling AE: StoreSCU	Dest Called AE: StoreSCP1

Host: *		Host: alpha
		Port: 1234

Note that in a Mapping Rule the maximum PDU lengths must be positive integers, and the destination port must also be an integer. The PDU Receive Lengths are not used when matching a mapping rule – they are used by the DICOM Switchboard to determine how big the "chunks" of data should be; the DICOM Switchboard will not read any PDU that has a length greater than 10 Megabytes.

Note that the asterisk (*) in the Source Calling AE and Source Called AE fields is *not* a wildcard as in a regular expression – it indicates that the Switchboard will match any value, but no "regex" processing is done to match AE titles.

Once the rule data is filled in, you may store it by clicking "*Update Rule*". Please note that all the fields must be filled in before the rule can be saved, and error messages will be reported if any data is missing or incorrect.

9.2.4 Manipulating Rules in a Group

Rules are listed and selected in the table in the top panel of the Mapping Rule Editor window.

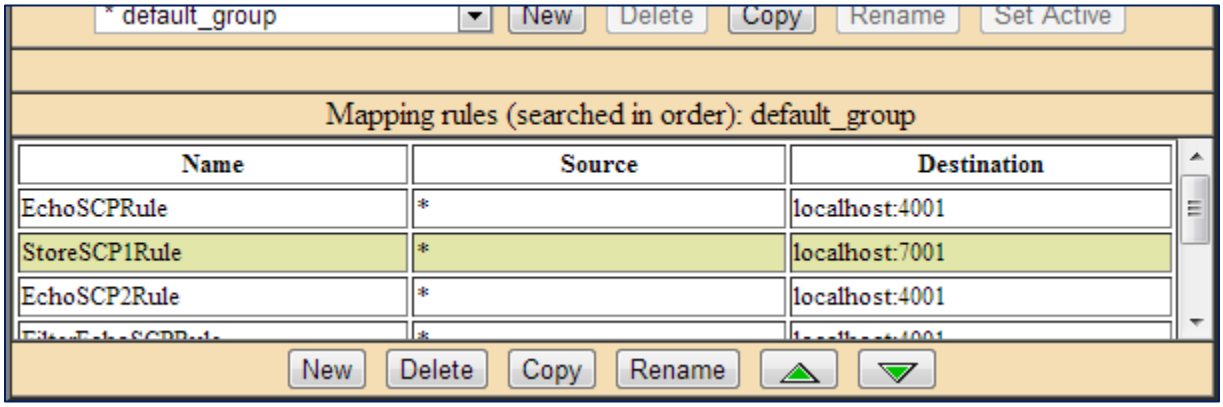


Figure 38 – Buttons for manipulating the rules in a group

The following options are available for manipulating rules:

- **Removing a rule** – To remove a mapping rule, select it in the table, then click the "*Delete*" button. You will be asked to confirm the deletion of the rule.
- **Copying a rule** – Select a rule and press the "*Copy*" button. You will be prompted to enter a name for the new rule; note that the name must be unique within the group of rules.
- **Renaming a rule** – Click the "*Rename*" button to change the name of the selected rule. You will be prompted to enter a new name for the rule, noting that the name must be unique within the group of rules.
- **Reordering the Rules** – The up and down arrow buttons  can be used to move rules up and down within the list of rules. Select a rule to be moved by clicking on it with the mouse pointer, then click on the appropriate arrow button to move the rule in the list.
  Why is the order of rules important? When the DICOM Switchboard looks for a matching

mapping rule to use, it searches rules in the order that they are listed; the first rule that matches is selected and applied. If your mapping rules are defined such that multiple rules could match, then the order of the rules is important and should be carefully considered. The more specific rules should be listed first and the less specific, more general, rules should be listed last.

For example, if you want messages for PrintSCP1 to go to host "alpha", while everything else is sent to host "beta", then you could use the wildcard (*) syntax in a rule to match "everything else", but you would want your PrintSCP1 rule for host "alpha" to be checked first by listing it first.

Remember: **Any changes you make to the mapping rules will *not* take effect until you click the "Save and Apply All Changes" button at the top of the window.**

9.2.5 Special Values in Mapping Rules

Each mapping rule has several fields that are used for matching incoming messages and determining what rule to use for forwarding the message. Some of the fields can have special values, including wildcards.

If a **source field** has a wildcard value of "*****" (asterisk, or star), then it will match any incoming value. For example, if the source Host is *, messages from any source will match. Similarly, using a Src Calling AE of * will match the calling AE-Title of any message.

(Note that the asterisk (*) in these fields is *not* a wildcard as in a regular expression – it indicates that the Switchboard will match any value, but no "regex" processing is done to match AE titles.)

Some **destination fields** can have the special value "**=**" (equals). You can set the Dest Calling AE and Dest Called AE to specific values, or you can use "=". If a destination value is set to "**=**", then the outgoing message will have the same value as the corresponding value in the incoming message. For example, if the Destination Calling AE Field is "**=**", then the destination's Dest Calling AE would be set to be the same as the source's Src Calling AE.

9.2.6 Schedules for Mapping Rules

As you create your mapping rules, you may wish to have some rules be applicable at certain times and not at others – for example, you may want the data to be sent to one destination during the work week but sent to another destination on weekends. One way to do this is to change your mapping rules before you leave work on Friday, but then you have to remember to change them back when you come in Monday morning. Fortunately, the Switchboard allows you to set schedules when rules are enabled.

By default, every rule is enabled all the time – you never have to wonder "when could this rule be applied?" However you can also specify a schedule for each rule (and not every rule needs to have a schedule), and it will be enabled only during those scheduled times. This could cause your set of mapping rules to look a little odd – you might think at first glance that you have duplicate mapping rules, rules with the same Source AEs and Destination AEs. But that is because you can say (for example) that StoreSCP1 goes to host alpha during the workday and then (via another rule, with a different schedule) that StoreSCP1 goes to host gamma at night.

To add a schedule to a mapping rule, you need to click the "Use Schedule" button near the bottom of the Source column and then click the adjacent **Edit** button.

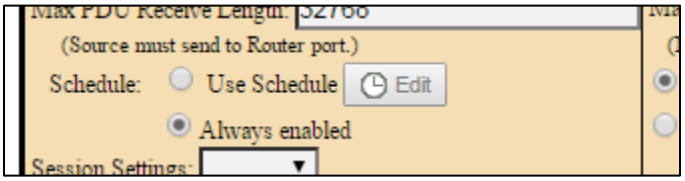


Figure 39 – Setting a schedule for a mapping rule

Clicking the Edit button will switch to a screen that allows you to choose the hours when the rule will be enabled (shown in **Figure 40** below).

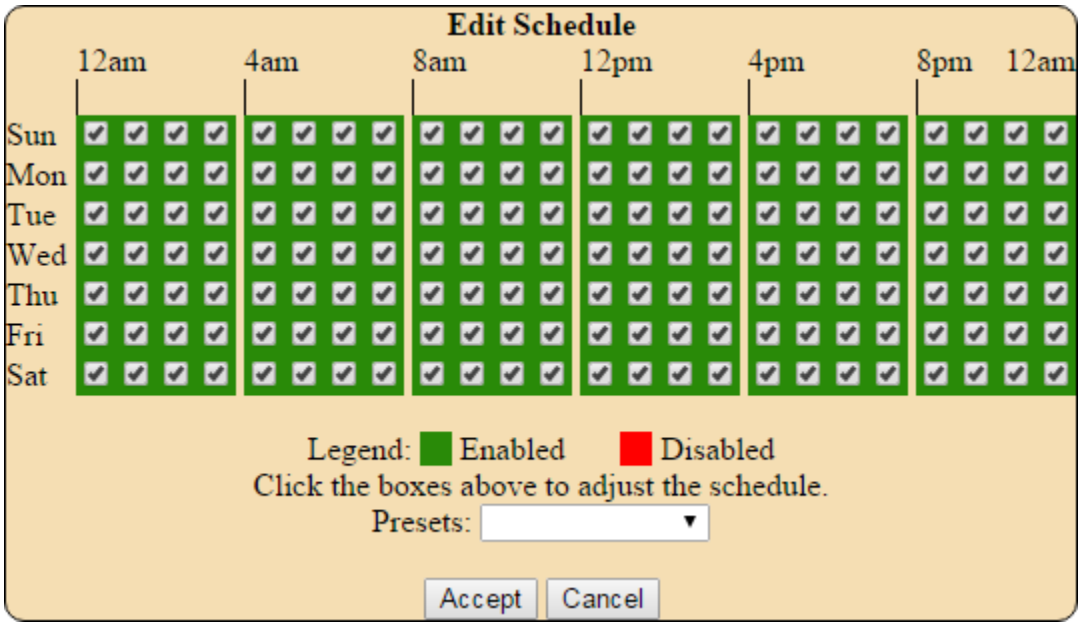


Figure 40 – Choosing the enabled hours

From this screen you can click the buttons representing each hour to enable or disable the rule during that time. Green indicates that the rule is enabled, red means that the rule is disabled. For example, **Figure 40** above shows that the rule is enabled 24 hours a day, 7 days a week, while **Figure 41** below shows that the rule will never run on Wednesdays.

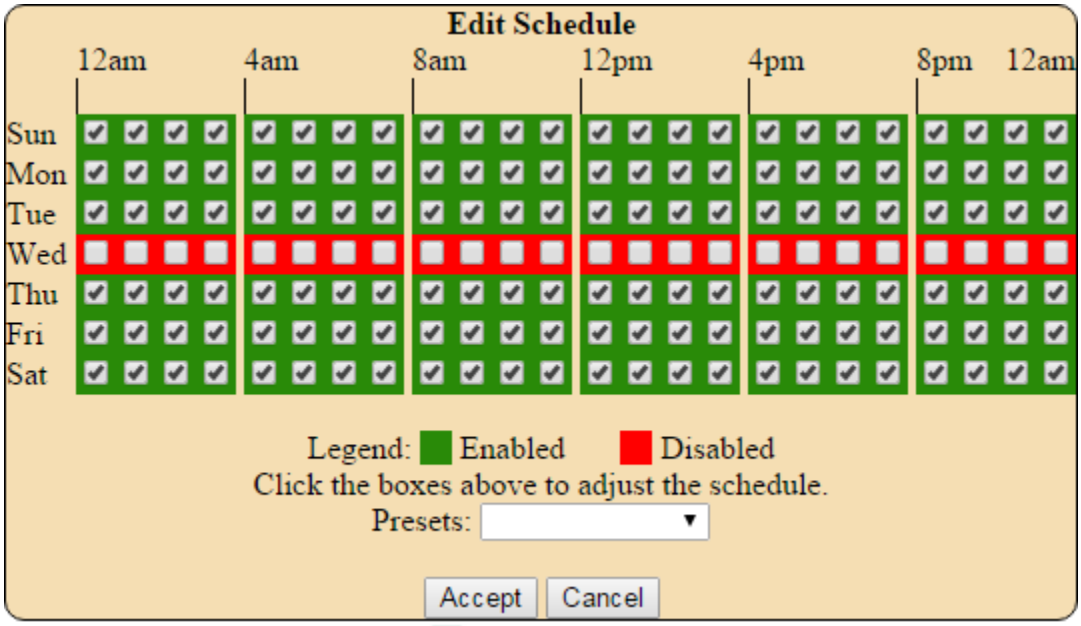


Figure 41 – A rule that never runs on Wednesdays

To make it easier to set a schedule for a rule, you can select from the list of presets at the bottom of the window and then modify the schedule as desired. For example, **Figure 42** below shows the list of presets and what the schedule looks like if a rule should be enabled only from 9am to 5pm on Monday through Friday.

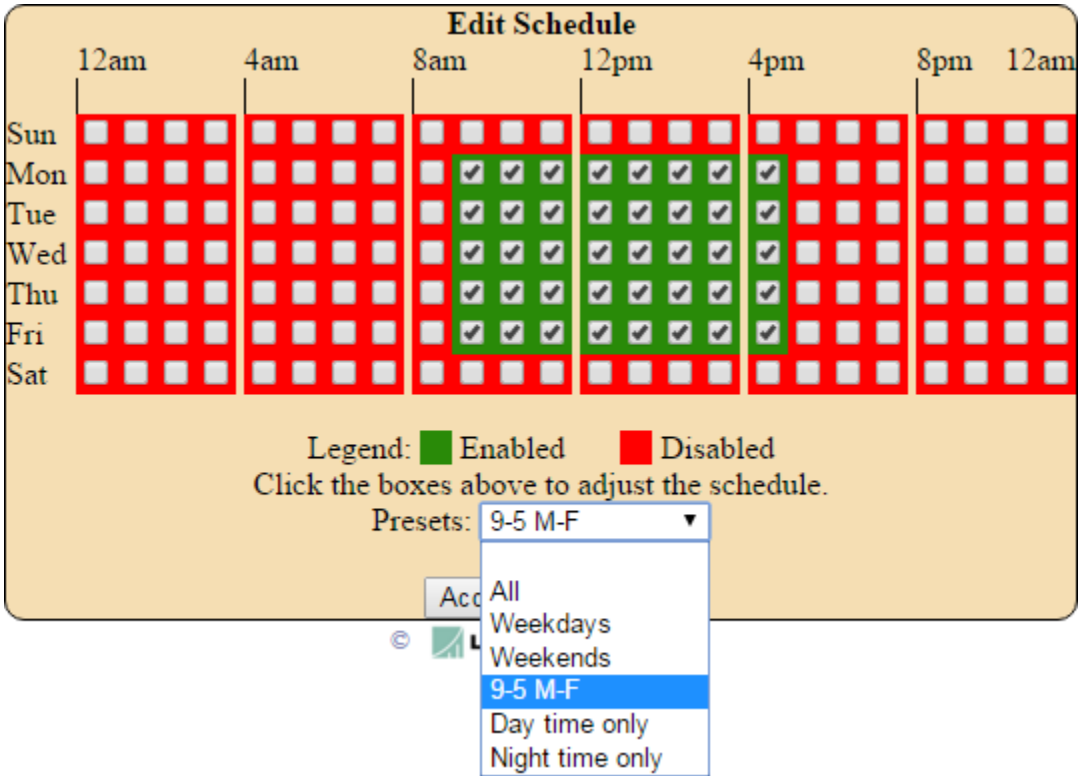


Figure 42 – The list of preset schedules

When you are done setting the schedule, click the Accept button; be sure to update the rule to store the changes and click "**Save and Apply All Changes**" to make the schedule changes take effect.

9.2.6.1 A Default Rule

Be careful when setting the schedules for mapping rules – it is not too difficult to forget to cover all the hours in a week, and then you would find that nothing goes through the Switchboard during those missed hours. For this reason, it is often useful to have a default rule with no schedule – this rule will be used when no other rule matches. (Due to the complexities of the mapping rules, it can make sense to have a default rule even if you don't have any schedules!) A default rule might have the following configuration:

Source	**DICOM Switchboard**	**Destination**
Src Calling AE: *****	Src Called AE: *	
	Dest Calling AE: =	Dest Called AE: =
Host: *****		Host: alpha
		Port: 1234

This means that anything that comes in, will go out with the same values but to your specified destination.
(Note: recall the meanings of the wildcard values * and =, described in Section **9.2.5** above.)

Please recall that mapping rules are searched in the order that they are listed. Such a default rule should be put last in the list – otherwise, you might find that none of your other mapping rules are being selected because the default rule is matched first.

9.2.7 Example Mapping Rules

The mapping rules determine how the SCUs and SCPs in a DICOM conversation communicate with each other through the DICOM Switchboard. To an SCU in a system with the DICOM Switchboard inline, the DICOM Switchboard acts as the SCP that the source SCU sends its messages to, and the Switchboard forwards those messages to the actual destination SCP. In this capacity, the actual destination SCP recognizes the DICOM Switchboard as the SCU and replies to the Switchboard. The DICOM Switchboard forwards any such replies from the destination SCP back to the source SCU.

Look at it this way:

[SCU] → [DICOM Switchboard] → [SCP]

The SCU communicates to the SCP via the DICOM Switchboard; the Switchboard acts as the SCP to the SCU and as the SCU to the SCP. The mapping rules tell the DICOM Switchboard where it should send the messages it receives – some can go to one place, and some can go to another.

Here are two example mapping rules along with an explanation of what they accomplish:

Example 1:

Name: **EchoSCPRule**	Rule to map ECHO SCP messages to the local box	
Source	**DICOM Switchboard**	**Destination**
Src Calling AE: *****	Src Called AE: ECHO_SCP	
	Dest Calling AE: **DCF_ECHO_SCU**	Dest Called AE: **ECHO_SCP**
Host: *****	Host: 0.0.0.0 (or localhost)	Host: **localhost**
Port: *****	Port: 2001	Port: **5019**
Max PDU Receive Length: **32768**	Max PDU Receive Length: 32768	Max PDU Receive Length: **32768**

This rule illustrates that any messages from any source host with any AE-Title requesting to talk to ECHO_SCP, will be sent on from the DICOM Switchboard to the destination with a Calling AE-Title of DCF_ECHO_SCU and a Called AE-Title of ECHO_SCP. (Note: the bold values are those that would typically be entered by a user configuring the DICOM Switchboard.) This setup also illustrates that the DICOM Switchboard has been configured to listen to port 2001.

Example 2:

Name: **StoreSCP1Rule**	Rule to map Store SCP messages to the local box	
Source	**DICOM Switchboard**	**Destination**
Src Calling AE: **StoreSCU**	Src Called AE: **StoreSCP1**	
	Dest Calling AE: =	Dest Called AE: =
Host: *	Host: 0.0.0.0 (or localhost)	Host: **localhost**
Port: *****	Port: 2001	Port: **6019**
Max PDU Receive Length: **32768**	Max PDU Receive Length: 32768	Max PDU Receive Length: **32768**

This rule illustrates that messages from any source host with an AE-Title of StoreSCU, desiring to talk to StoreSCP1, will be forwarded on to the server on localhost listening on port 6019. Because the "=" (equal) semantics are used here for the Dest Calling and Called AE, the forwarded messages will use the same Calling AE and Called AE as the incoming message.

Put another way, a message from StoreSCU comes into the DICOM Switchboard, requesting the StoreSCP1 service. The DICOM Switchboard finds those values in a mapping rule and knows to forward the message on, acting as the SCU (with AE-Title StoreSCU) and requesting StoreSCP1 from the actual destination SCP.

Please note that the PDU Receive Lengths are not used for matching on a rule; they are used by the DICOM Switchboard to determine how big the "chunks" of data should be. In addition, the DICOM Switchboard will *not* read any PDU that has a length greater than 10 Megabytes. *Also, when the DICOM Switchboard is operating in its default or normal "transparent" mode, then the negotiation for max PDU size is handled end-to-end between the source and destination, and the Switchboard internal values are effectively ignored.*

Also note that on the source side of the configuration you should not normally specify a source port – the port that the data is coming from *will* vary from one association to the next and is set by the

source system, however the destination listening port won't change with associations and should be set to match the configuration of the destination.

9.2.8 Operational Modes: Transparent vs. Filtering

Transparent mode is the behavior of the DICOM Switchboard where PDUs are read from the source socket and regenerated on the destination socket with no changes to the PDU payload.

A note about transparency: Transparent Mode provides transparency from the DICOM perspective. It does not provide transparency from the network perspective. Because DICOM PDUs are regenerated and forwarded by the DICOM Switchboard, the IP address reported by the protocol for forwarded packets will not be the address of the original source device. An example of where this might be relevant is when some system enforces access rules based on the Calling AE Title and the source IP address. Such a device will not recognize packets forwarded by the DICOM Switchboard as legitimate. To accommodate such a case, the user should add the Calling AE Title and IP address of the DICOM Switchboard to that system's list of recognized devices. Once this is done, forwarded communication should be accepted by the destination device.

As was also noted above, when the DICOM Switchboard is operating in "transparent" mode, the negotiation for maximum PDU size is handled end-to-end between the source and destination, and the Switchboard's internal values are effectively ignored.

Filter mode is the behavior of the DICOM Switchboard where PDUs read from the source are converted into DIMSE messages by the Switchboard, optionally filtered, and then converted back into DIMSE messages and the associated PDUs before being forwarded on to the destination.

Filter mode has the advantage of being able to alter DIMSE messages and re-package PDUs into different sizes. Transfer syntaxes can also be altered in this mode. Some disadvantages of filter mode are that there is some (small) overhead with decoding/re-encoding each message and that problems relating to decoding/encoding of datasets might be either masked by the DICOM Switchboard or introduced by the DICOM Switchboard. Masking a problem may be considered an advantage if it allows two devices that could not previously communicate to then begin communicating; however in a test environment, it may be important to know that two devices that can communicate through the DICOM Switchboard cannot communicate directly.

Note: Some of the statistics used by the Status Monitor are accumulated only for mapping rules that are operating in Filtered mode. This is because Transparent mode does not parse the PDU data but simply forwards it. One such statistic is the DIMSE message counter; if this Status Monitor counter never changes, but messages are successfully passing through the DICOM Switchboard, it means that all of the mapping rules are using Transparent mode.

9.2.9 Setting the Filters for a mapping rule to use

If a mapping rule uses Filter mode, you should attach filters to the rule to check and/or modify the PDU data by selecting filter sets from the drop-down lists at the bottom of the Mapping Rule Editor's display.

- Use the "**Filters: To Dest**>>>" list to choose the filters for data going from the source to the destination. (This is the filtering that you usually want to do.)
- Use the "**<<<Filters: To Source"** list to choose the filters for data going from the destination back to the source.

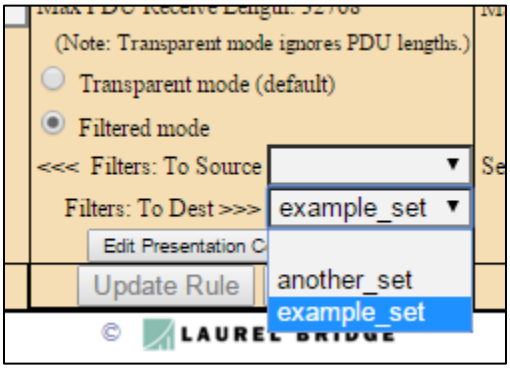


Figure 43 – Lists for selecting the filters in a mapping rule

Select the filter set to use from the drop-down list. Typically you select one of the global sets (note: these are stored in a common location, usually *"*`$DCF_CFG\dicom\filter_sets`*"*) so that they can be used by multiple applications. Please also note that global sets can only be selected here - you must select "*Edit Global Filter Sets*" from the Switchboard Operations Window to edit global sets.

9.2.10 Mapping Presentation Contexts

Filtered mode also allows the DICOM Switchboard to map presentation contexts by modifying the requested presentation contexts before forwarding them to the accepter. This allows you to change how data is transmitted through the DICOM Switchboard. You can change the abstract syntax, the list of requested transfer syntaxes, the role negotiation items, and the sop specific data. For example, by mapping input data with ILE transfer syntax to a JPEG compression transfer syntax, it is possible to have uncompressed ILE image data be both forwarded to the destination and compressed automatically as it flows through the DICOM Switchboard.

You can add or edit the presentation context mappings by clicking the "**Edit Presentation Context Mappings"** button at the bottom of the center panel. The screen will change to an editor window where you can modify how the presentation contexts are mapped, especially specifying what transfer syntaxes to accept and to request. This editor screen is shown **below**.

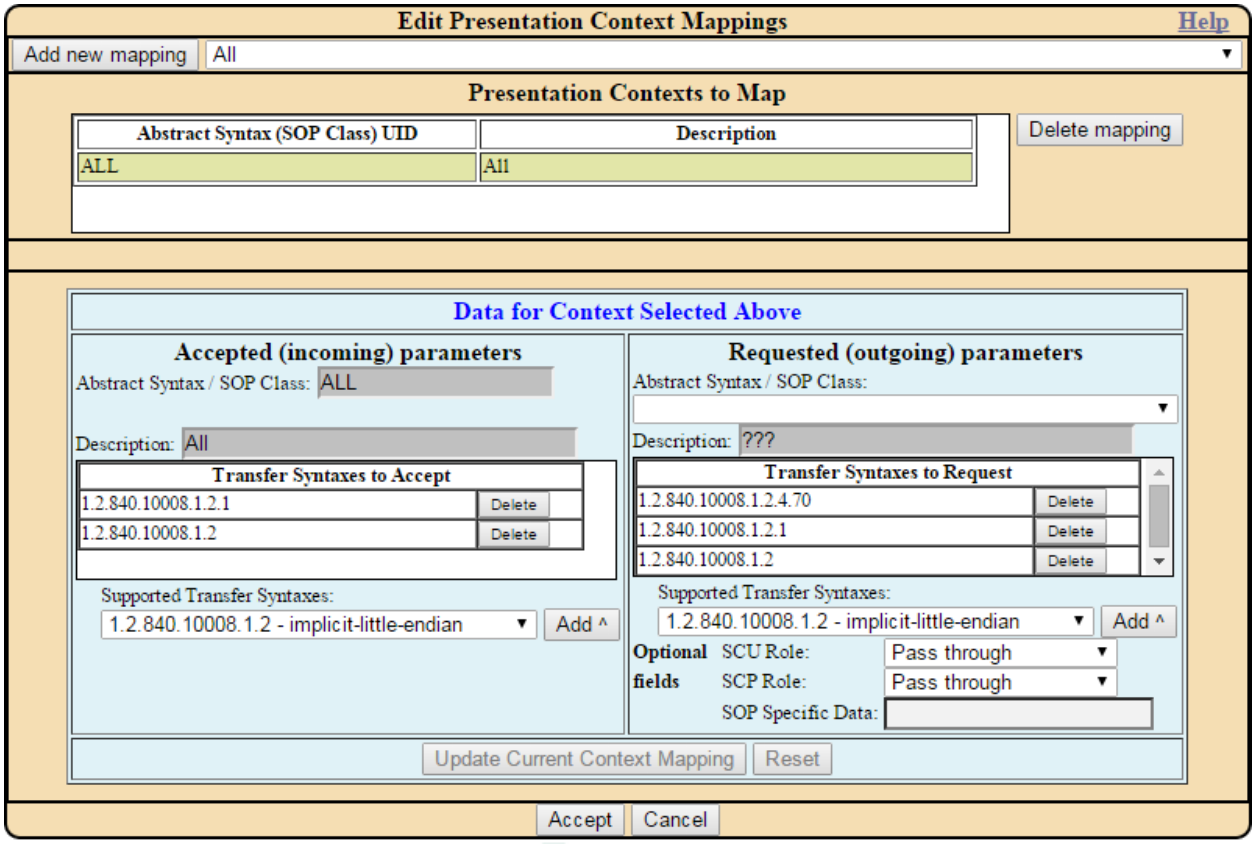


Figure 44 – Modifying the Presentation Context Mappings

You may choose to map a specific presentation context – any that do not match will be processed without any changes, and the association will proceed unchanged. Alternately, you may choose to map "All", causing your mapping to be applied to all presentation contexts. The table at the top lists the presentation context mappings for the currently selected mapping rule.

The example shown **above** will map any Abstract Syntax (SOP class), and it will accept and convert the data from Explicit Little Endian (ELE) or Implicit Little Endian (ILE) to JPEG Lossless if the destination supports it. If the JPEG option is not accepted, then it will try Explicit Little Endian (ELE) and Implicit Little Endian (ILE), in that order.

Presentation mapping configuration options:

- **Adding a new mapping** – Select the UID of the presentation context (the SOP class UID) that you wish to map, or "ALL" to map any presentation context. Then click the "*Add new mapping*" button at the top-left of the editor GUI. Once you have chosen the presentation context to map, you may edit it in the panel below the table.
- **Deleting a mapping** – Select the mapping you wish to delete in the table at the top. Click the "*Delete*" button, and then confirm that you wish to delete the mapping.
- **Editing a mapping** – When you select a context mapping in the table at the top, its related information is displayed in the fields in the lower half of the window. You may modify these values as you desire. Note that you *must* specify the "Transfer Syntaxes to Accept" and the "Transfer Syntaxes to Request". Also note that the order of the syntaxes in the tables is important since DICOM application entities process the request and select the first acceptable option, based on the order they were presented. A drop-down list for the table entries allows

you to select any of the transfer syntaxes supported by the DICOM Switchboard's configuration.

You may also change the SCU role or SCP role selection data. The values (1 or 0) for "Propose" and "Do Not Propose" are as defined in Appendix D.3.3.4 in Chapter 7 (Message Exchange) of the DICOM standard. The option "Pass through" will allow the existing values to pass through unchanged; select "Do Not Pass through" to prevent the values from being transmitted to the destination/accepter.

You may also change the outgoing Abstract Syntax to whatever you want; if you don't specify an outgoing Abstract Syntax (by leaving the field blank), the same one that was accepted will be used for the outgoing connection.

1. **Finishing an edit session** – When you are done changing the data for the presentation context mapping, click the "*Update Current Context Mapping*" button at the bottom to store the changes in the mapping rule. ***Note*** that you must still save the mapping rule itself (in the main GUI) in order to save the changes to the configuration database used by the application. If you make a mistake when editing a context mapping, you may click "*Reset*" to revert to the original data, unless you have already clicked "*Update*".

   Click the "*Accept*" button when you are finished editing the presentation context mappings for the mapping rule. This will return you to the mapping rule editor GUI – remember to store the updated mapping rule and to commit your changes! (You may be warned if changes have been made but not stored.) Or you can click "*Cancel*" to discard any changes you have made and return to the main Mapping Rule Editor.

Note on Recompressing Images:

You can sometimes use a Presentation Context Mapping to correct errors in the compression of an image that prevent it from being viewed – for example, if the Huffman Encoding Tables for JPEG have invalid coefficients (see the *DCF Release Notes* for *DCF 3.3.58c*), you can configure Switchboard to recompress the data and correct the tables (recompression is best used only with **lossless** syntaxes). To do this, set up your Presentation Context Mapping to accept one Transfer Syntax on the inbound side and then to request the same Transfer Syntax on the outbound side (recall that Presentation Context Mappings can only be used in Filtered Mode). Then you must do the following *manual* configuration steps:

1. Stop Switchboard.
2. Edit the file `<INSTDIR>/cfg/apps/defaults/dcf_switch` with a text editor such as Notepad or VIM.
3. Find the setting for `enable_compression_pass_through_mode`, in the section "`java_lib/DCS/default_session_cfg`". By default, its value is "yes"; change it to be "**no**".
4. Save your changes to the file.
5. Start Switchboard.

9.2.11 Examples of Presentation Context Mappings

9.2.11.1 Example: Force image data to be compressed:

Suppose you have CT images and MRI images that are being stored through the DICOM Switchboard to an archive, but you want the CT images to be forwarded compressed as JPEGs, while you want

the MRIs images to be stored just as they were originally transmitted. Further suppose that you already have a mapping rule for (let us say) StoreSCP1 so that the DICOM images are redirected properly through the DICOM Switchboard to your destination archive. Let's examine how you can also set up a Presentation Context Mapping to modify how the two types of data are transmitted.

You can customize your mapping rule for StoreSCP1 to set up a presentation context mapping for only the CT Image Storage SOP class that would accept a certain set of transfer syntaxes and request a different set, starting with a JPEG Compression transfer syntax. In this scenario any associations for the StoreSCP1 mapping rule that request the CT Image Storage SOP class will have their data compressed using JPEG compression. Since you didn't set up a context mapping for the MRI Image Storage SOP class, the MRI data will be passed through unmodified by any context mapping.

This table that follows shows what the required Presentation Context mapping rule might look like:

Map **CT Image Storage – 1.2.840.10008.5.1.4.1.1.2** Abstract Syntax (SOP Class)	
Accepted (incoming) parameters	**Requested (outgoing) parameters**
Abstract Syntax / SOP Class: 1.2.840.10008.5.1.4.1.1.2	**Abstract Syntax / SOP Class:**
Description: CT Image Storage	**Description**:
Transfer Syntaxes to Accept	**Transfer Syntaxes to Request**
1.2.840.10008.1.2.1 – Explicit VR Little Endian	1.2.840.10008.1.2.4.70 – JPEG Lossless First Order Prediction
1.2.840.10008.1.2 – Implicit VR Little Endian	1.2.840.10008.1.2.1 – Explicit VR Little Endian
	1.2.840.10008.1.2 – Implicit VR Little Endian

Notes:

6. The order of the transfer syntaxes to request is important, since they will be requested in that order and the first one that is accepted is the one that will be used.
7. The outgoing Abstract Syntax is blank; if you don't specify an outgoing Abstract Syntax, the same one that was accepted will be used for the outgoing connection.

An important note on JPEG compression: When you compress images to JPEG Baseline Process 1 (1.2.840.10008.1.2.4.50) or any other lossy compression, the result is a new image (being lossy compressed, the result image will not be exactly identical to the source image); a new image requires a new SOP Instance UID, according to the DICOM standard. In such a case, the Switchboard will automatically create a new SOP Instance UID for the image and add to the dataset any necessary derived fields to show that the image was compressed at some point in time. (As you are testing transmission of images through the Switchboard, this could make it seem like you are getting duplicate images when you send the same image multiple times – each new image will have a new SOP Instance UID.)

One way to avoid the creation of new UIDs is to use JPEG .70 compression (1.2.840.10008.1.2.4.70 – JPEG Lossless First Order Prediction). JPEG .70 is a lossless compression – since the resulting *image* will be identical to the source *image*, a new UID is not required. To avoid the creation of the derived fields, you should manually modify the configuration file for dcf_switch ("`$DCF_CFG/apps/defaults/dcf_switch`" [$DCF_CFG is the "cfg" directory under the Switchboard's installation directory, e.g., "`C:\LB Switchboard\cfg`" – see **Figure 6**). Search the file for "jpeg_lossless", and set the following configuration values:

- `add_derived_image_fields_for_mono` → false
- `add_derived_image_fields_for_color` → false

Save the changes and then restart the DICOM Switchboard.

Note: See **Appendix C: RLE Transfer Syntax Support** for information on using RLE compression in the Switchboard.

An important note on Jpeg2000 lossy compression: If you want to use Jpeg2000 lossy compression (1.2.840.10008.1.2.4.91), the default value for fractional bits (or "fracbits") is 12. You can also use 10 or 13 for the fracbits value. To use a different value, you must manually edit the dcf_switch configuration file ("`$DCF_CFG/apps/defaults/dcf_switch`"). Search the file for "**`DCS/DicomTSCWCodec/plugins/TSCWJasper`**" and change the DLL base name attributes:

```
[ DCS/DicomTSCWCodec/plugins/TSCWJasper ]
# Windows Jasper plugin dll with fracbits=12
# Plugins are available for fracbits = 10, 12, and 13.
win_dll_base_name = DCF_TSCWJasper12
# Linux Jasper plugin dll with fracbits=12
# Plugins are available for fracbits = 10, 12, and 13.
unix_dll_base_name = DCF_tscwjasper12
```

Save the changes and then restart the DICOM Switchboard. (See the DCF Release Notes for 3.3.50c for a detailed explanation of this.)

9.2.11.2 Example: Force image data to be uncompressed:

Suppose you know that your SCU will request an association with a JPEG compression transfer syntax, but you do *not* want **any** of the data to be compressed when it arrives at the destination, perhaps because you have a destination that cannot support JPEG compressed data. To handle this situation, you could set up the list of transfer syntaxes to accept to include JPEG compression, but set the list of transfer syntaxes to request to be the common syntaxes of ILE, ELE, and EBE. In this case, the SCU could still request JPEG compression and the DICOM Switchboard would accept, but the data would be transmitted uncompressed from the DICOM Switchboard, without you having to modify the SCU or SCP behavior.

The required Presentation Context mapping might look like the following:

Map **ALL** Abstract Syntaxes (SOP Classes)	
Accepted (incoming) parameters	**Requested (outgoing) parameters**
Abstract Syntax / SOP Class: ALL	**Abstract Syntax / SOP Class:**
Description: Match any SOP class / abstract syntax	**Description:**
Transfer Syntaxes to Accept	**Transfer Syntaxes to Request**
1.2.840.10008.1.2.4.70 – JPEG Lossless First Order Prediction	1.2.840.10008.1.2.1 – Explicit VR Little Endian
1.2.840.10008.1.2.1 – Explicit VR Little Endian	1.2.840.10008.1.2 – Implicit VR Little Endian
1.2.840.10008.1.2 – Implicit VR Little Endian	1.2.840.10008.1.2.2 – Explicit VR Big Endian

Note again that the outgoing Abstract Syntax is blank; if you don't specify an outgoing Abstract Syntax, the same one that was accepted will be used for the outgoing connection.

9.2.11.3 Example: Forwarding from an unsupported modality:

Finally, let's say you will be forwarding MG images from a Mammography device to an archive that does not support that modality. You already know what filtering you need in order to make a C-Store-Request for an MG image look like a CR image. The problem is that if the MG presentation context is not accepted, you will never get a chance to perform the DIMSE message filtering you have configured. In this case you can add a presentation context mapping that will convert the requested MG contexts received from the SCU to CR contexts that will be forwarded to the SCP during association negotiation.

For the sop-class/abstract-syntax UID string, we select MAMMO-FOR-PRESENTATION (1.2.840.10008.5.1.4.1.1.1.2). Next choose as the outbound sop-class COMPUTED-RADIOGRAPHY (1.2.840.10008.5.1.4.1.1.1). Select the set of accepted transfer syntaxes (i.e., those that will be accepted from the SCU) and the set of requested transfer syntaxes (i.e., those that will be requested of the SCP). If the remote SCP accepts the CR context, then the DICOM Switchboard will indicate the acceptance of the MG context back to the SCU. DIMSE messages will then be routed between the two devices on that context using the appropriate transfer syntax and filtering configuration.

Your Presentation Context mapping rule might look something like this:

Map **1.2.840.10008.5.1.4.1.1.1.2** – Digital Mammography X-Ray Store – For Presentation	
Accepted (incoming) parameters	**Requested (outgoing) parameters**
Abstract Syntax / SOP Class: 1.2.840.10008.5.1.4.1.1.1.2	**Abstract Syntax / SOP Class:** 1.2.840.10008.5.1.4.1.1.1
Description: Digital Mammography X-Ray Store – For Presentation	**Description:** Computed Radiography Image Storage
Transfer Syntaxes to Accept	**Transfer Syntaxes to Request**
1.2.840.10008.1.2.1 – Explicit VR Little Endian	1.2.840.10008.1.2.1 – Explicit VR Little Endian
1.2.840.10008.1.2.2 – Explicit VR Big Endian	1.2.840.10008.1.2.2 – Explicit VR Big Endian
1.2.840.10008.1.2 – Implicit VR Little Endian	1.2.840.10008.1.2 – Implicit VR Little Endian

Reminder: If you edit the filters or the presentation context mappings, be sure to save the updated mapping rule, or your changes may be lost!

9.3 Testing the Mapping Rules

As you create your mapping rules, you will probably find that making them do what you want is not always easy – you may expect that rule X would be used for a certain modality, only to find that rule Y is being used instead. This is because the rules must be matched correctly, and the order of the rules is also important. The Switchboard's **Mapping Rule Tester** is a simple utility designed to help you make sure that the desired mapping rule is chosen when given certain inputs. For example, you might find that you need to reorder your rules so that a rule with many wildcards is checked last.

The DICOM Switchboard finds the mapping rule to use based on the source modality's hostname (or IP address), Src Calling AE, and Src Called AE, and – optionally – the time and day. You can enter these values into the Mapping Rule Tester, and it will simulate the connection and tell you what

mapping rule will be used for the values you input. This approach allows you to adjust and to tweak your mapping rules without the requirement of sending actual data from your modalities.

The Mapping Rule Tester is a separate application that is launched from the Windows Start menu:

`Start → All Programs → Laurel Bridge Software → Switchboard → Utilities → Run Mapping Rule Tester`

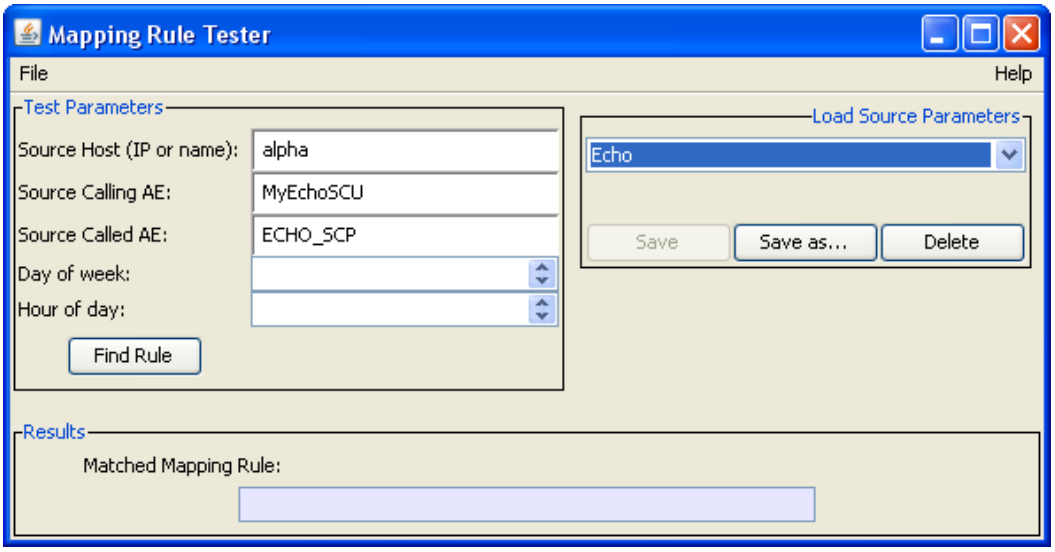


Figure 45 – Testing a Mapping Rule

For Linux: The Mapping Rule Tester may be launched from a Switchboard Linux Command Terminal by running the command "`bin/Run_MappingRuleTester.bash`".

To use the Mapping Rule Tester, fill in the "**Test Parameters**" fields with the values used by a modality that will be contacting the DICOM Switchboard.

- **Source Host (IP or name)** – the hostname or IP address of the source modality that will be sending to the DICOM Switchboard
- **Source Calling AE** – the Calling AE-Title of the source modality
- **Source Called AE** – the AE-Title being called by the source modality

These fields are matched by the DICOM Switchboard to choose the mapping rule that will be used. You must specify all of these fields in order for a mapping rule to be chosen. **Note** that wildcards (such as *, or star) are not allowed – a wildcard value would not *in actuality* be sent from your modality to the DICOM Switchboard, a real value would be sent.

You can also test any schedules that a mapping rule may have by setting the **Day of week** and the **Hour of day** fields. If they are blank, then the current day and time will be used. By setting these fields, you can test if the desired mapping rule is selected when your schedule is in effect. For example, you can test if your mapping rule is chosen at 3:00AM on a Thursday without actually having to stay up until 3:00AM to test it.

After the data is entered, then when you click the "*Find Rule*" button, the name of the mapping rule that would be used for those values will be displayed at the bottom.

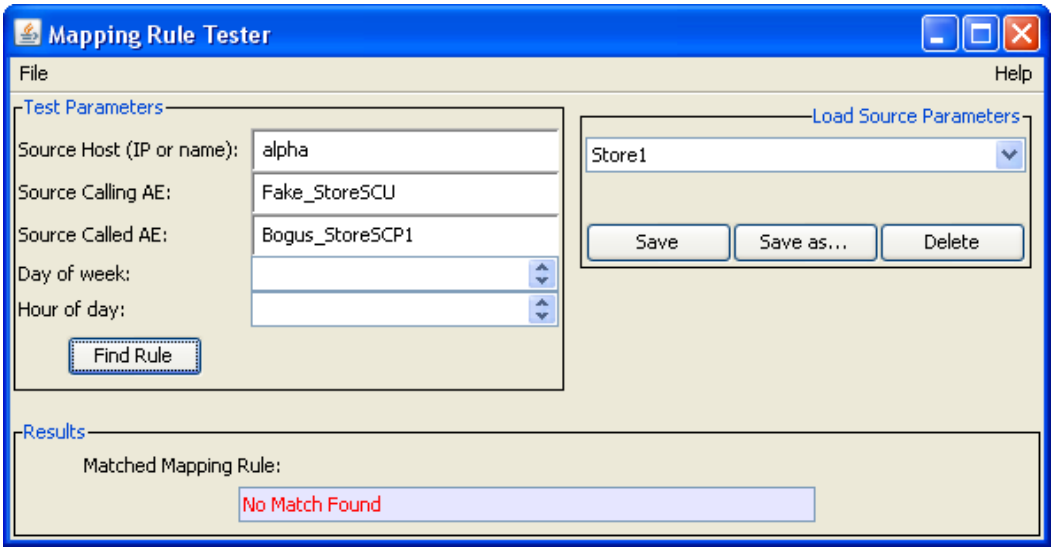


Figure 46 – A mapping rule without a match

9.3.1 Reusing Test Parameters

As you enter test parameters and see what mapping rules are selected, you may find yourself entering the same values over and over again. The Mapping Rule Tester allows you to save sets of test parameters and reuse them, selecting them from the drop-down list at the right, under "**Load Source Parameters**". When you select a set of source parameters from the drop-down list, the set's values are automatically copied into the "Test Parameters" fields. This permits you to test with them again and again.

You may save the test parameters you use by clicking the appropriate buttons below the list. Click "*Save as…*" to create a new entry in the list; you will be prompted for a name. Click "*Save*" to store the present test values in the currently selected entry in the list. You may also remove the currently selected entry by clicking the "*Delete*" button.

Note: As you test your Switchboard's configuration, you may wish to clear your destination SCP's database or cache periodically. This is because users often test the Switchboard by sending the same image multiple times. In many cases, the receiving SCP will correctly receive the image the first time; after that, it will think that it already has the image and so report success but it will not have actually received the image – it will report success but just throw the incoming image away. If this happens, you may not see the desired AE title or filter changes.

10 Switchboard Status Monitor

The DICOM Switchboard Status Monitor shows the state of the DICOM Switchboard and the messages passing through it. The "lights" and counters displayed on it will change when a connection's status changes (for example, is closed) or as a value changes. This allows you to conveniently see that data is going through the DICOM Switchboard and if any errors may be occurring in the connections. When the Switchboard Status Monitor is used in conjunction with your source device, it allows you to see at a glance that data is passing through the DICOM Switchboard.

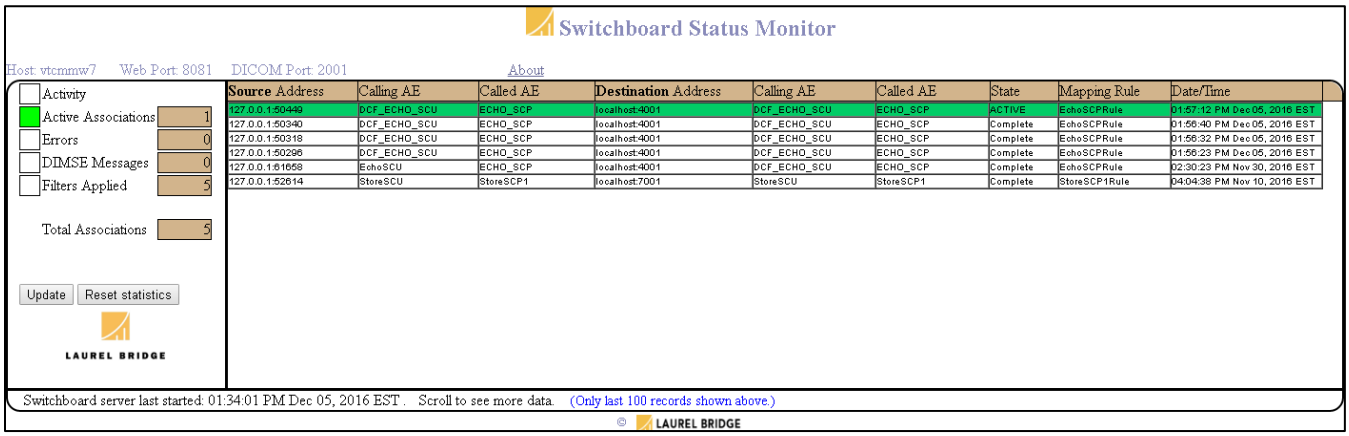


Figure 47 – Switchboard Status Monitor Screenshot

The table shows information about the connections to the DICOM Switchboard. This includes the source, destination, and Calling and Called AEs of the clients connecting via the DICOM Switchboard. It also shows the name of the Mapping Rule that was used – this can help you to know that your Mapping Rules are being matched correctly. (Note that the Status Monitor only remembers that last 100 transactions – remembering more would slow down the Status Monitor and use a lot of memory. As the table grows, the oldest rows [those at the bottom] are removed when the list grows too long.)

The counters show the current values of the desired information: active associations, errors, and DIMSE messages, as well as the total number of associations.

Please note that the DIMSE message counter only changes for mapping rules that are in Filter Mode; Transparent mode cannot count the DIMSE messages since they are never decoded. If this counter never changes, it means that all the mapping rules are in Transparent mode (which is the default).

You may press the "*Reset statistics*" button to clear the current counts and any inactive connection information. Counters are also reset when the DICOM Switchboard is re-started.

The Status Monitor periodically polls the Switchboard framework for the latest connection information. Alternatively, pressing the "*Update*" button will immediately request the latest data from the DICOM Switchboard.

If you are not seeing any activity on the Status Monitor, you should check that the mapping rules are correct and that your clients are configured to send to the host **and** port for the DICOM Switchboard. You should also verify that the destination is configured to accept associations from the Switchboard's hostname, port, and AE-Title.

Note: When you are configuring the DICOM Switchboard (as illustrated in Chapter **7, Configuring the DICOM Switchboard** above), it is possible to disable the collection of the connection statistics.

In such a case, the connections table will not be updated; instead, you will see a warning message and instructions on how to re-enable those statistics.

Note: As of version 3.3.40c, the Switchboard uses a non-applet version of the Status Monitor that is more browser-friendly. However, the old Java version is still available and may be used in a non-applet mode.

For Linux: The Status Monitor may also be launched from a Switchboard Linux Command Terminal by running the command "`bin/Run_StatusMonitor.bash`".

Note: On Windows the Status Monitor (Java app) may be run via the command "`Run_StatusMonitor.bat`" from a Switchboard Command Prompt.

11 Logging and the Real-Time Log

Besides filtering DICOM data, the DICOM Switchboard can be used to monitor DICOM associations and the DICOM messages sent between SCUs and SCPs. As the data on the association is being filtered; it is sometimes helpful to monitor and observe these transactions. As the DICOM Switchboard forwards the protocol messages, it can also log information about the associations and the messages. By default a low level of logging is used, but this level can be adjusted in real-time without restarting any of the participants in the associations. Selecting the "*Set Logging Verbosity*" link from the Switchboard Operations page will allow you to see the different levels and kinds of logging data that the DICOM Switchboard can provide. The screenshot below illustrates the available settings.

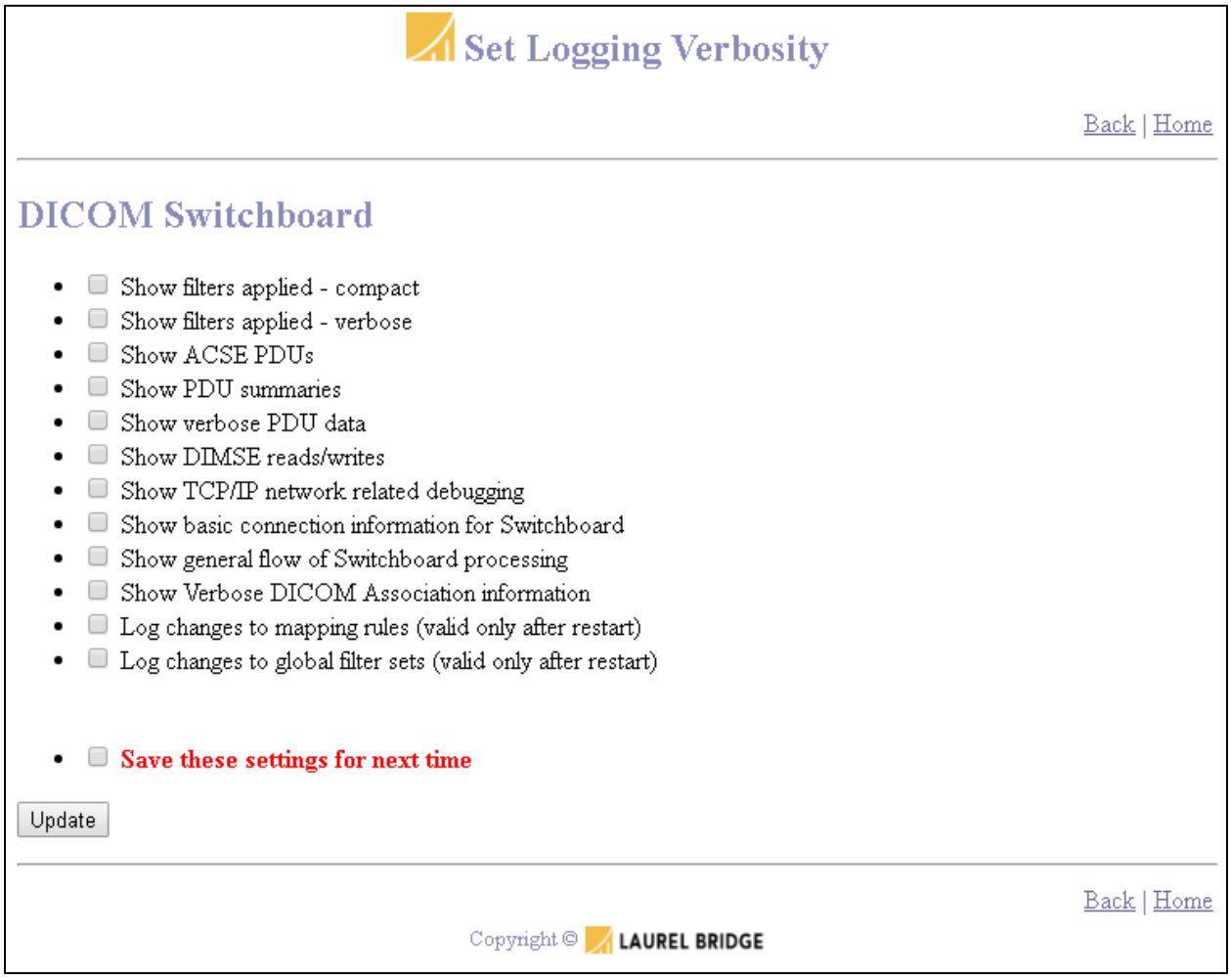


Figure 48 – Adjusting the Logging Verbosity

These logging levels are roughly in increasing order of the amount of information they provide. (Please note that some of these levels – such as "Show verbose PDU data" – can provide a *lot* of data and may slow the operation of the DICOM Switchboard. Verbose logging should be used with caution!) If the box "Save these settings for next time" is left unchecked, only the currently running processes will be affected and will adjust their logging verbosity appropriately; check this box if you wish for the new logging level to be in effect the next time the DICOM Switchboard is started. You may adjust the logging verbosity by checking or unchecking the boxes as you desire; when you are done, press the "*Update*" button to save the new logging settings.

If you wish to view the logging data from the DICOM Switchboard in real-time as it occurs, click "*View Real-Time Log*" on the Logging tab. This will start a web application in a new window. The web app receives data that is logged to the DICOM Switchboard and its components and then displays that data in a scrolling pane as it is received. The "activity lights" on the sides of the display will flash as data is received to let you know that data is coming in.

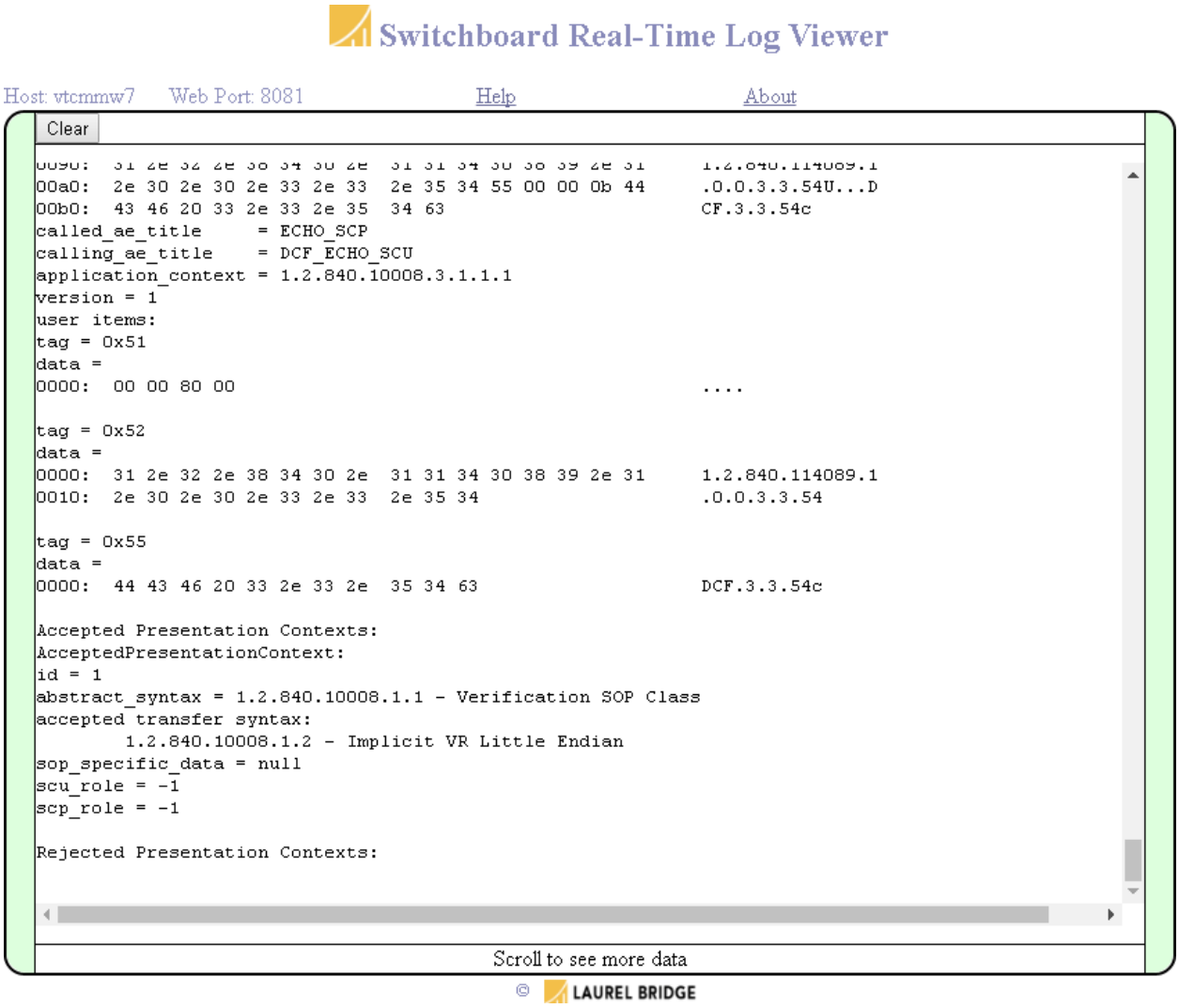


Figure 49 – DCF Real-Time Log Screenshot

Text is scrolled off and out of the display area when the amount of text becomes too great. However, the full log data may still be viewed by clicking "*View Log Files*" (on the Logging tab of the Switchboard Operations page) and selecting a log file to display; the "`dcf_switch`" and "`system.log`" files will contain the information on transactions being processed by the DICOM Switchboard. (An example listing of log files is shown **below**.) Please note that *very* verbose data may slow both the Real-Time Log viewer and the DICOM Switchboard.

> **For Linux:** The Real-Time Log Viewer (Java app) may also be launched from a Switchboard Linux Command Terminal by running the command

"`bin/Run_RealTimeLog.bash`". **Note** that you may need to edit the bash script and change the port number to be whatever TCP port you have configured the Log Server to use.

Figure 50 – Some of the log files generated by the DICOM Switchboard

Because the amount of data in the log files can grow dramatically, especially with a high level of logging verbosity, there are times you will wish to clear the log files. This is easily done from the Switchboard Operations page using the "*Clear Log Files*" link on the Logging tab. However, there may be times when you wish to keep the system log file for further analysis in case an error occurs. In that case, you may save the log file: When you are viewing a log file, clicking the "*Save log file*" link at the top of that page will copy the file and timestamp the filename – a window will open up telling you the location and name of the archived log file.

To keep the log files from filling up your disk, the DICOM Switchboard uses "rotating log files". The way this works is as follows: the DICOM Switchboard permits a certain number of log files for each server to exist; when that number is exceeded, the oldest one is deleted. You may configure the number of log files allowed, as well as the maximum size of each log file, from the "Configuring the DICOM Switchboard" page (see Chapter 7). The default number of log files is 5, and a new log file is started when an existing one tops 3MB; you may permit unlimited files and/or unlimited size by setting these values to "-1".

11.1 Debugging a Switchboard in Slave mode

If you have multiple Switchboards that are synchronized to one "Master" Switchboard (see Section **7.1 Synchronizing Multiple Switchboards** for how to do this), it can be hard to debug a problem on one of the "slave" Switchboards. That is because the slave boxes will keep their debug flags in sync with the Master – only when the debug flags on the Master are changed will the debug flags on the slave change, and even then *all* the slave boxes will be changed to match!

To diagnose a connection problem on just one slave box, you will need to enable the debug flags on just that one box, diagnose and fix the problem, and then disable the debug flags.

To enable the debug flags on slave box:

1. Go to the Configuration tag on the Switchboard Operations page and click "Configure Switchboard".
2. On the Configuration page, uncheck the "Run as Slave" box under DCDS Server Configuration.

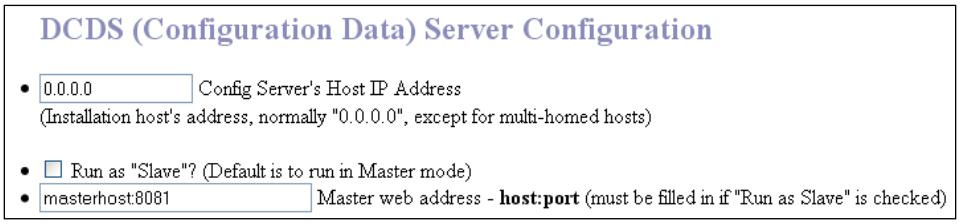


Figure 51 – Changing to Master from Slave mode.

3. Click the "Update" button at the bottom of the page and then click "Home" to return to the Switchboard Operations page.
4. Stop and then Start the Switchboard from the Operations page. (The "slave" box is now no longer in slave mode and can be modified independently of the Master and other Slaves.)
5. Enable the debugging flags by clicking on the Logging tag and then "Set Logging Verbosity", as described at the beginning of Chapter **11**.

Now you can debug the connection problems by increasing the verbosity of the logging, sending connections through the Switchboard, and diagnosing problems that are seen in the log files.

Reverse the process when you are done to return the computer to slave mode.

1. Click "Configure Switchboard" on the Configuration tab.
2. Check the "Run as Slave" box and then click "Update". (The Master's web address should not need updating.)
3. Stop and then Start the Switchboard from the Operations page. The box you just diagnosed will be in slave mode again, and its mapping rules, filters, and debug flags will sync up with the master.

12 Utilities

Various utility programs are included with the DICOM Switchboard. These are primarily useful for debugging and testing situations. The available utilities are described below.

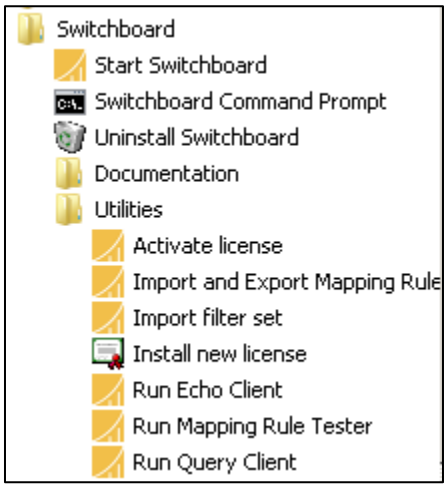


Figure 52 – Accessing the DICOM Switchboard's Utilities via the Start Menu

12.1 DICOM Echo Clients (Echo SCU)

As you use the DICOM Switchboard and configure it, you will encounter problems where you are not sure if data is going through the DICOM Switchboard as you expect. One way to check the communication through the DICOM Switchboard is to send an "echo" (DICOM Verification) message through it. The DICOM Echo SCU tests connectivity with another DICOM entity by sending a DICOM Echo message (also known as a DICOM Verification message) to that entity and then listening for a response message back from the destination host.

The DICOM Switchboard includes some simple echo clients that you can use to test the Switchboard's communication and configuration. You may use these in addition to whatever echo capabilities your own modalities possess.

12.1.1 Java-based Echo GUI

There is a simple Java-based GUI for testing echo; this is accessed via the Windows Start menu as:

```
Start → All Programs → Laurel Bridge Software → Switchboard →
Utilities → Run Echo Client
```

This will bring up a simple GUI where you may specify the AE titles and destination of the echo message. Simply enter the appropriate networking configuration and press the "Send DICOM Echo" button to perform the test. Note that in a default DICOM Switchboard configuration, the default values for this GUI will send to the Switchboard's built-in Echo Server by going through the DICOM Switchboard.

(**For Linux**: Open a command terminal to the DICOM Switchboard's installation directory, set its environment by "dotting" the dcf.env file, and run the command `./bin/JEcho_SCU.bash`.

Press the "*Send DICOM Echo*" button to send the echo request. An error will be displayed if the echo fails.

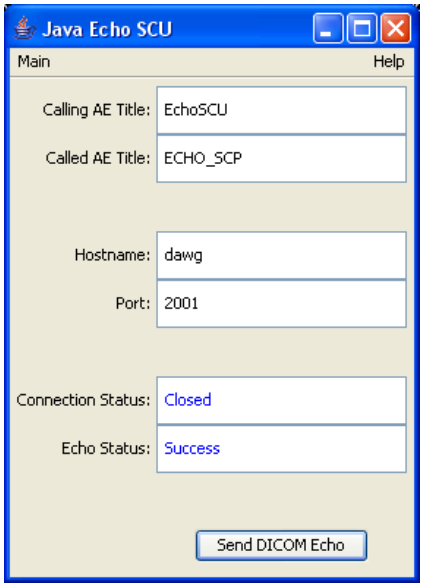


Figure 53 – Java Echo SCU GUI

12.1.2 Command-line Echo

The DICOM Switchboard also includes a command-line echo client; this should be run from a Switchboard Command Prompt (or Switchboard Linux Command Terminal) so that the system environment settings are correctly set:

```
Start → All Programs → Laurel Bridge Software → Switchboard →
Switchboard Command Prompt
```

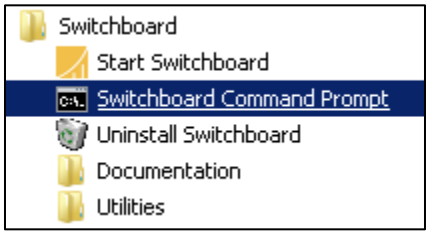


Figure 54 – Accessing the Switchboard Command Prompt

The echo client is named "**dcf_echo_scu**"; you may see its usage by specifying "-h" as a command line option, for example:

```
dcf_echo_scu -h
```

To use it to test the basic connectivity of the DICOM Switchboard in its default configuration, run the command:

```
dcf_echo_scu localhost 2001
```

This will send an echo message to the DICOM Switchboard, which will forward the echo to the Switchboard's built-in echo server. (Note: specify the echo SCP listening port that you chose when you configured the DICOM Switchboard; the Switchboard's Echo SCP uses port 4001 by default.)

12.1.3 Testing the DICOM Switchboard using DICOM Echo

Another way to test the communication through the DICOM Switchboard is to launch the Switchboard Status Monitor and then to send an echo request through the DICOM Switchboard to the Echo Server (do this by using one of the methods described above to send to the echo server described below). If the DICOM Switchboard is configured correctly, you will see the data on the Status Monitor change. If there is an error and the echo does not complete, you may need to adjust your configuration settings and try again.

12.2 DICOM Echo SCP

The Switchboard's DICOM **Echo SCP** provides access to a DICOM Echo server (also known as a "DICOM Verification" server).

When you start the DICOM Switchboard, a simple echo (verification) SCP is started as well, to allow you to test the Switchboard's communications. This Echo SCP uses port 4001 by default on Windows, but you can change this on the Switchboard Configuration page, accessed from the Switchboard Operations page.

The Switchboard's default set of mapping rules includes one rule that will route echo requests with a Called AE Title of "ECHO_SCP" to this built-in echo server.

12.3 DICOM Query/Retrieve SCU

The DICOM **Query/Retrieve SCU** allows the user to perform DICOM queries to another DICOM entity. These queries can be configured to request any information which can be retrieved via DICOM tags.

As you use the DICOM Switchboard, you may find that you need to query one of your clients or servers for DICOM data to determine how to configure some of the Switchboard's settings. For example, you may want to see how an archive is storing some data that has been sent to it so that you can adjust your filters appropriately.

The DICOM Switchboard includes two simple Query-Retrieve (Q/R) clients to help you with this task.

12.3.1 Java-based Query-Retrieve (Q/R) GUI

The first is a simple Java-based GUI; this application is accessed via the Windows Start menu as:

```
Start → All Programs → Laurel Bridge Software → Switchboard → Utilities
→ Run Query Client
```

(**For Linux**: Open a command terminal to the Switchboard's installation directory, set its environment by "dotting" the dcf.env file, and run the command `./bin/JQuery_SCU.bash`.)

This application allows you to select the server that you want to query and to see the data records that match your query. It has an advanced configuration option that lets you modify the DICOM elements to query and to display.

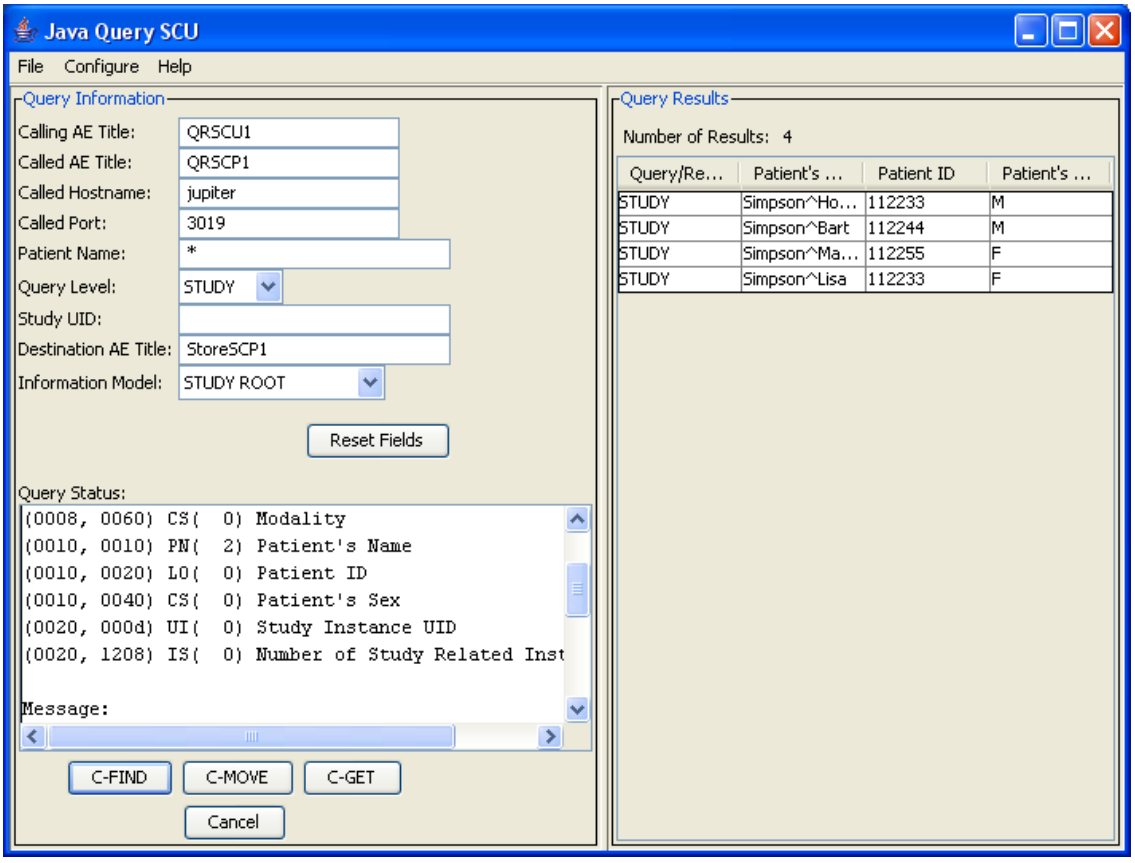


Figure 55 – Java Query SCU GUI

12.3.2 Command-line Query-Retrieve (Q/R) Client

There is also a command-line Q/R client: dcf_qr_scu. This application should be run from a Switchboard Command Prompt (or Switchboard Linux Command Terminal) so that the system environments setting are correctly set:

```
Start → All Programs → Laurel Bridge Software → Switchboard →
Switchboard Command Prompt
```

The GUI version of the Q/R client is simpler to use, but the command-line version is more powerful. See the application docs for dcf_qru_scu for more information on how to use it, or run the command with the "-help" option as:

```
dcf_qr_scu -help
```

12.4 DICOM Store SCP

The built-in DICOM **Store SCP** provides access to a DICOM store server for use by the DICOM Switchboard for testing and support of integration tasks.

One of the most common uses of the DICOM Switchboard is to modify data as it is being stored from one device to another – for whatever reason, the unmodified data may not be accepted by the store server you are using, so you wish to use the DICOM Switchboard to modify it so that it is acceptable. The DICOM Switchboard provides a *simple* store server that can be used for testing purposes – you can modify the Switchboard's mapping rules to route data to this store SCP and then examine the

data that is saved to verify that it was modified as you expected. If the data received is not acceptable, you can modify your filters and resend the data until the data is acceptable. This allows you to test that your filters are correct without sending to the *actual* store server and then having incorrect data on it.

To use the built-in Store SCP:

1. Configure the DICOM Switchboard so that it will automatically start the Store SCP when the Switchboard is started.
    a. From the Switchboard Operations Window, select the "Configure Switchboard" option.
    b. At the bottom of the page are the configuration options for the Test Store Server. If necessary, modify the port that it will use. (The port defaults to 7001, but you can make this port whatever you want, as long as another application is not using the port you select.)

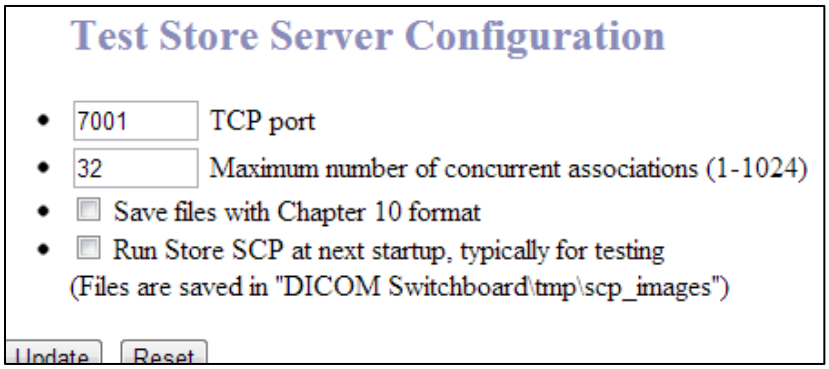


Figure 56 – Configuring the Test Store SCP

    Note: You will need to remember this value when you configure the mapping rules to send to the Test Store Server.
    c. Check the box adjacent to **"Run Store SCP at next startup, typically for testing"**. This will tell the DICOM Switchboard to start the built-in Store Server as part of its normal startup process.
    d. The Store SCP – as a simple example – by default will save the files in non-Chapter 10 format. Click the checkbox next to "**Save files with Chapter 10 format**" if you want any data to be saved in Chapter 10 format.
    e. Click "*Update*" to save the new configuration settings.
2. Configure your mapping rules to send the desired store jobs to the sample Store SCP. Under default settings this would be to host: localhost, port: 7001.
3. Restart the DICOM Switchboard so that the new mapping rules take effect and the Store SCP is running.
4. Test the filters by sending store jobs through the DICOM Switchboard into the Store SCP, modifying your filters as necessary.
5. When you are done testing with the store server, go to the "*Configure Switchboard*" page and uncheck the option "*Start Store SCP for testing next time*", and click "*Update*".
6. Reconfigure your mapping rules to send to your production store SCP, and restart the DICOM Switchboard.

12.4.1 Testing the DICOM Switchboard using DICOM Store Client

The DICOM Switchboard includes a simple Store SCU to send images for storage. You can test the DICOM Switchboard and the filters you have created by storing an image through the Switchboard and checking the results.

- First, configure the DICOM Switchboard to start the Store Server, as described above in steps 1 through 3 – *remember* to restart the DICOM Switchboard so that the Store Server is running.
- Then open a Switchboard Command Prompt (or Linux Command Terminal) and change to the "test/images" directory.

  ```
  Start → All Programs → Laurel Bridge Software → Switchboard →
  Switchboard Command Prompt
  ```

- Start the Status Monitor so you can observe the transaction as it happens.
- Send the test image through the DICOM Switchboard and to the Store Server with this command:

  ```
  dcf_store_scu -c StoreSCP1 localhost 2001 test.dcm
  ```

- Examine the file that was saved to make sure it was filtered appropriately – this is described below.

The files saved by the sample store SCP are put in the "tmp/scp_images" subdirectory under the directory where you installed the DICOM Switchboard, typically,

```
C:\LB Switchboard\tmp\scp_images.
```

(Note: for **Linux**, this directory is typically like `/opt/DSB-3.5.2c/tmp/scp_images`.) See **Figure 6** to see the Switchboard's directory structure.

One way to examine the result images to see if the data was filtered as you desired would be to use the Switchboard's Dump File application. Open a Switchboard Command Prompt from the Windows Start Menu, change to the image directory, locate the file of interest and run

```
dcf_dump.exe <filename>
```

To see the options for this utility program, run `dcf_dump.exe -h`; this causes the help info to be displayed.

As you use the DICOM Switchboard to filter DICOM data and to monitor transactions and diagnose connectivity problems, you may at times find it useful to use the Switchboard's simple store client (`dcf_store_scu`) to send images through the Switchboard or directly from the Switchboard's host to your destination. For example, this could help you to diagnose any connectivity problems that may occasionally occur with the DICOM Switchboard.

12.5 Import Filter Set

This option, which is available from the Windows Start Menu, provides a simple way of adding existing filter sets to the DICOM Switchboard for its use. (See **Appendix B: Section 1** for more information.)

12.6 Displaying a File's Data Set

It is often helpful to be able to view the contents of a DICOM dataset. The DICOM Switchboard provides two utilities that can assist you in this task.

One such app – **dcf_dump** – is used to display the dataset in a DICOM file. To use it, open a Switchboard Command Prompt (or Linux Switchboard Command Terminal) from the Windows Start Menu and specify a DICOM file as an argument to the application, e.g.,

```
dcf_dump my_dicom_file.dcm
```

To see the options for this utility program, run `dcf_dump.exe -h`; this causes the help information to be displayed.

The DICOM Switchboard also provides a web interface to this utility program, it is accessible via the link "*Display a file's data set*" on the Switchboard Operations window. When you select this option the following interface is displayed:

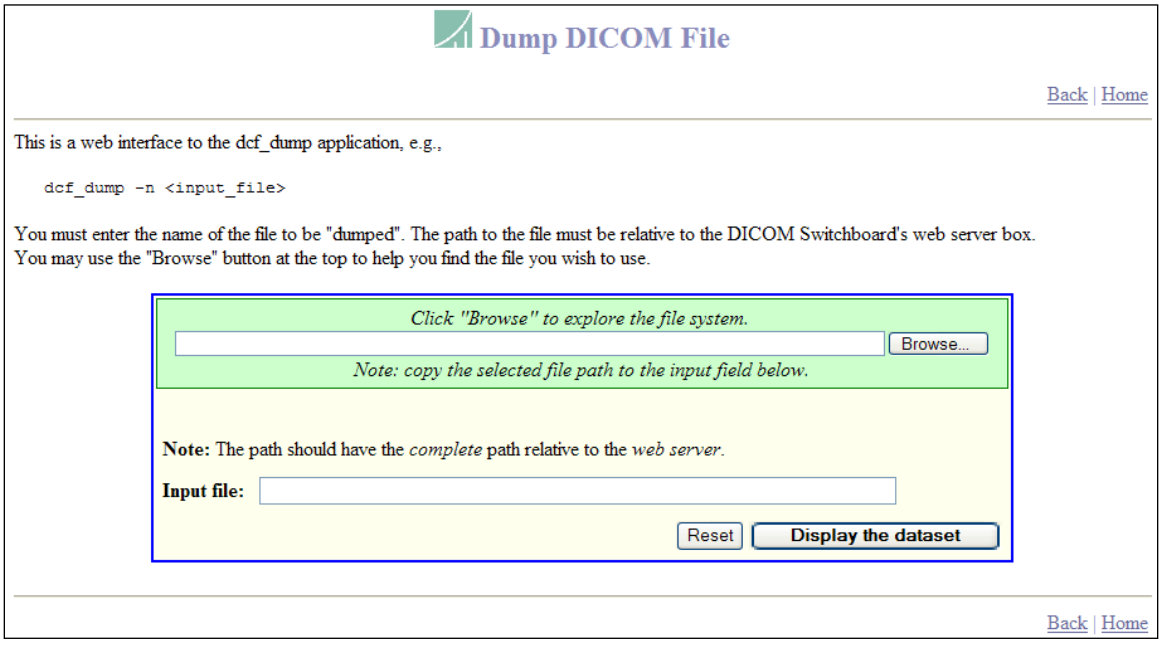


Figure 57 – Web interface to Dump a DICOM file

You can enter the name of a DICOM file that you wish to "dump" and view the DICOM dataset in your browser. Note that the path to the file in this GUI is relative to the Switchboard's web server host. (For Internet Explorer, it may be necessary to precede the file's path with a backslash '\'.) You can use the "Browse" button to explore the local file system to help you find the file to dump – then copy that filename into the input file field.

12.7 Manually Filtering a DICOM File

Another application that is included with the DICOM Switchboard is a simple command-line application for filtering the data set in a DICOM file. This application – **dcf_filter** – takes a configuration file that specifies the filters to apply and allows you to set the input data object to apply the filters to, as well as the output file for the filtered data. (You could also use this utility to test your filters to make sure they operate as intended.)

To use this utility, open a Switchboard Command Prompt (or Linux Command Terminal) from the Windows Start Menu and execute the application as illustrated below:

```
dcf_filter -f <filter> -i <inputfile> -o <outputfile>
```

To see the options for this utility program, run `dcf_filter.exe -h`; this causes the help info to be displayed.

The DICOM Switchboard also provides a web interface to this program; you access it via the link "*Filter a DICOM file*" on the Switchboard Operations window.

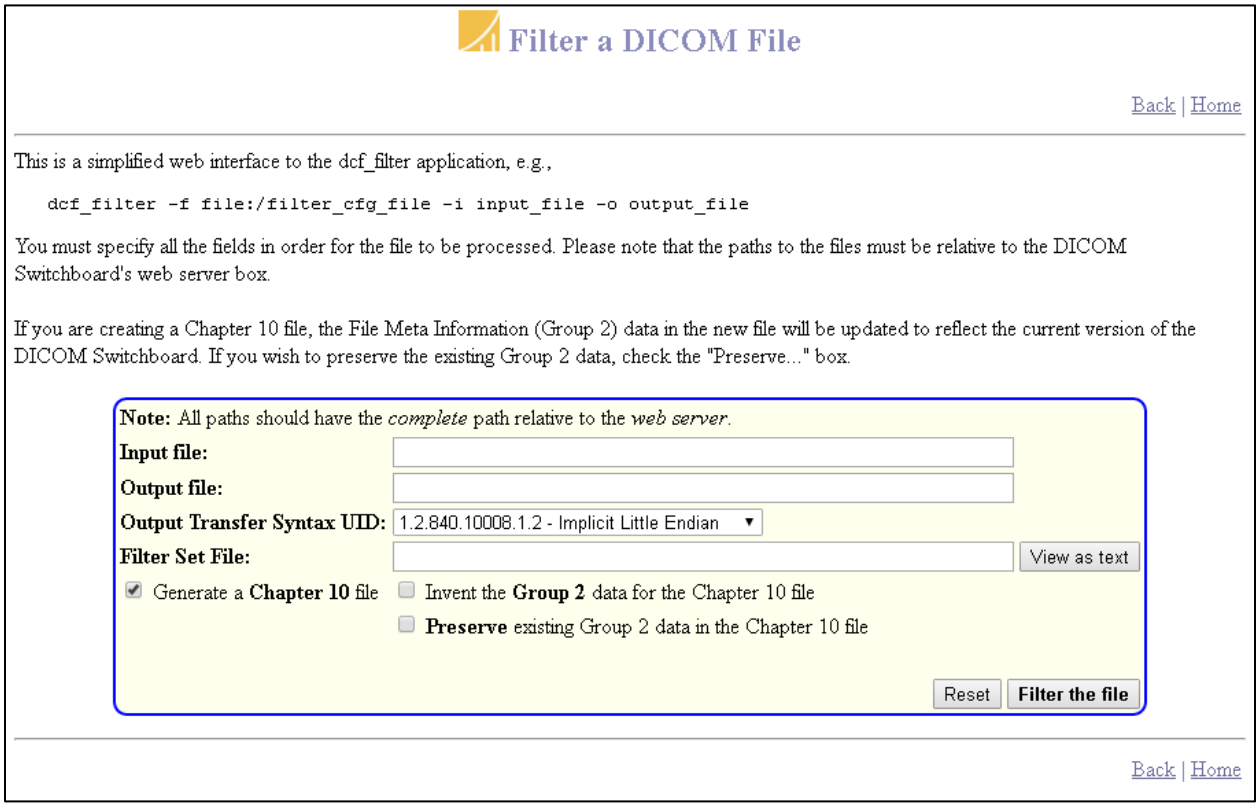


Figure 58 – dcf_filter web interface

You can enter the name of the DICOM file that you wish to filter, specify the output file and the output's transfer syntax, and enter the name of the filter set file that you wish to use. When you click the "*Filter the file*" button, the results of the filtering operation are shown in your web browser.

You may use the "Browse" button in the lower left-hand corner to explore the file system to find the files that you wish to sue. (For Internet Explorer, it may be necessary to precede the files' paths with a backslash '\'.) Find the file you wish to use, and then copy the value into the desired input field.

Since you are creating a new DICOM file, dcf_filter will automatically update the File Meta Information (Group 2 elements) to reflect the current version of the DICOM Switchboard. If you wish to preserve the existing Group 2 elements, check the "Preserve" box.

12.8 Mapping Rule Tester

The Switchboard's **Mapping Rule Tester** is a utility designed to help you make sure that the desired mapping rule is chosen when given certain inputs. Since the DICOM Switchboard finds the mapping rule to use based on the source modality's hostname (or IP address), Src Calling AE, and Src Called AE, this utility allows you to enter these values and then simulates the connection to tell you what mapping rule will be selected and used for the values you input. This approach allows you to adjust your mapping rules without the requirement of sending actual data to and from actual modalities.

See Section **9.3, Testing the Mapping Rules** for detailed information on using the Mapping Rule Tester utility.

12.9 Import and Export Mapping Rules

This application provides a simple method of importing new sets of mapping rules into the DICOM Switchboard, as well as a method for exporting the existing sets of mapping rules. Importing mapping rules could be useful, for example, if you have upgraded to a newer version of the DICOM Switchboard and want to import your old mapping rule groups. This could also be useful if you have multiple systems with Switchboards installed on them and you want all of them to use the same mapping rules – you could create the mapping rules on one computer, export them, and then import them onto the other computers, instead of creating the rules anew on each computer.

This application is designed to work on the **groups** of mapping rules in the DICOM Switchboard, not on individual mapping rules.

When you launch the Importer/Exporter, you will be given the choice of importing or of exporting mapping rule sets.

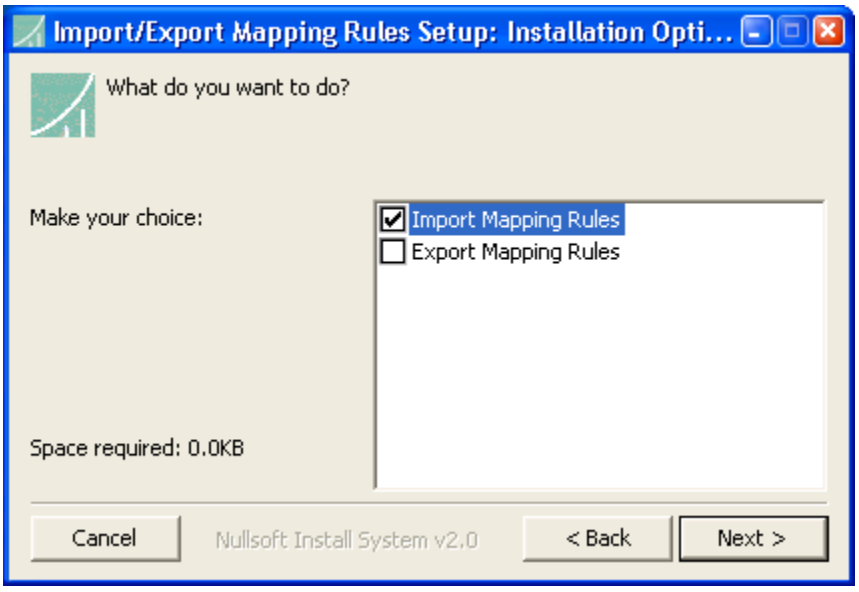


Figure 59 – The Mapping Rule Importer and Exporter

You launch the Importer/Exporter application from the Windows Start menu as follows:

`Start → All Programs → Laurel Bridge Software → Switchboard → Utilities → Import and Export Mapping Rules`

12.9.1 Exporting Mapping Rules

Launch the Importer/Exporter application from the Windows Start menu, choose the Export option, and then click "*Next*". You will be asked to specify the name of the file where the mapping rule groups should be exported. The file of exported rules will be created in the Switchboard's installation directory (typically, "`C:\LB Switchboard`"). Note that a file with the selected name cannot already exist.

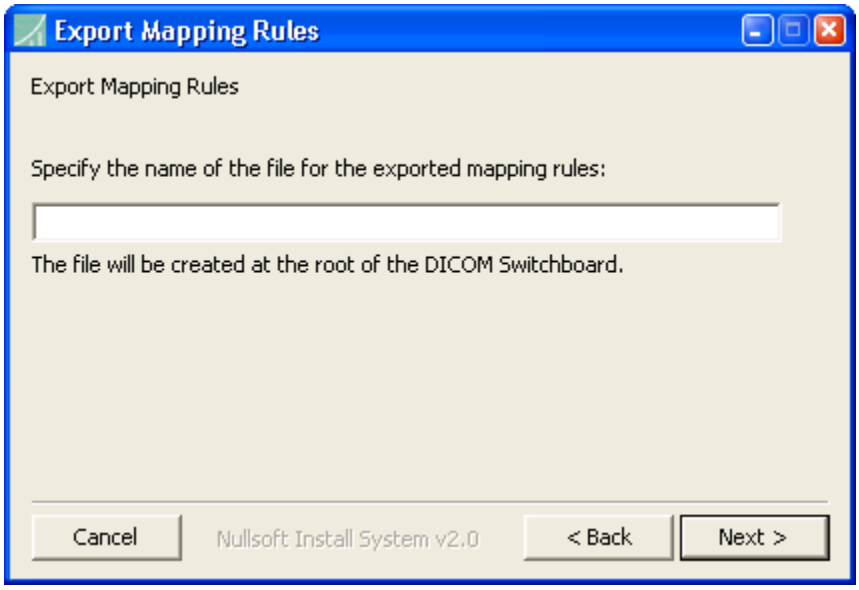


Figure 60 – Choose the file to export the mapping rules to

If the operation succeeds, the results will be shown to you, like the example **below**.

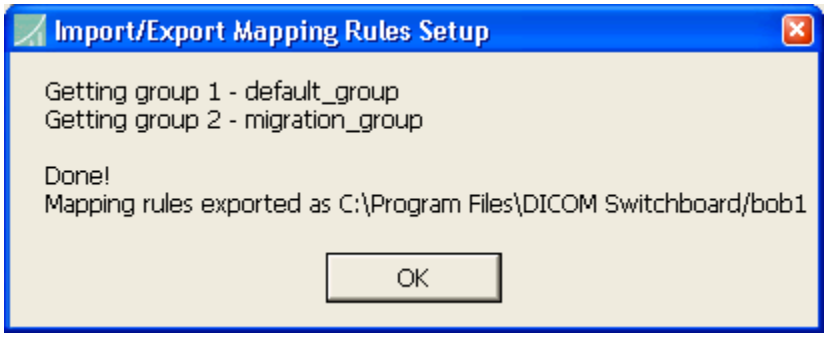


Figure 61 – Results of exporting mapping rules

For Linux: From a Switchboard Linux Command Terminal, run the command

```
perl export_mapping_rules.pl
```

and follow the prompts. (See **Figure 64** below.)

12.9.2 Importing Mapping Rules

Launch the Importer/Exporter application from the Windows Start menu, choose the Import option and click "*Next*". You will be asked to specify the name of the file with the mapping rule groups that you wish to import.

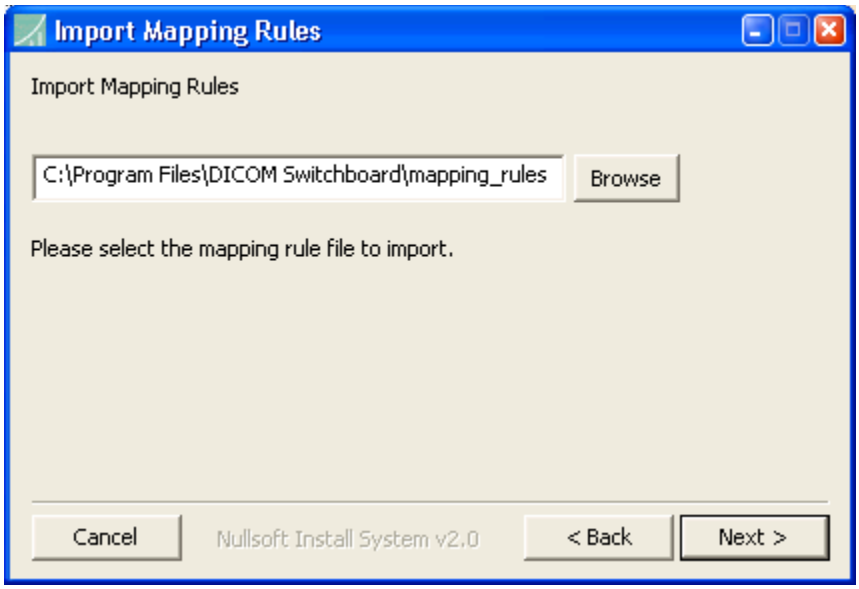


Figure 62 – Choosing the file of mapping rules to import

If any of the groups that you are trying to import has the same name as a set of mapping rules in your DICOM Switchboard, the imported set will be renamed by adding a numbered suffix to it. The results (as shown **below**) will tell you the new name (if required) of each group of mapping rules that was imported.

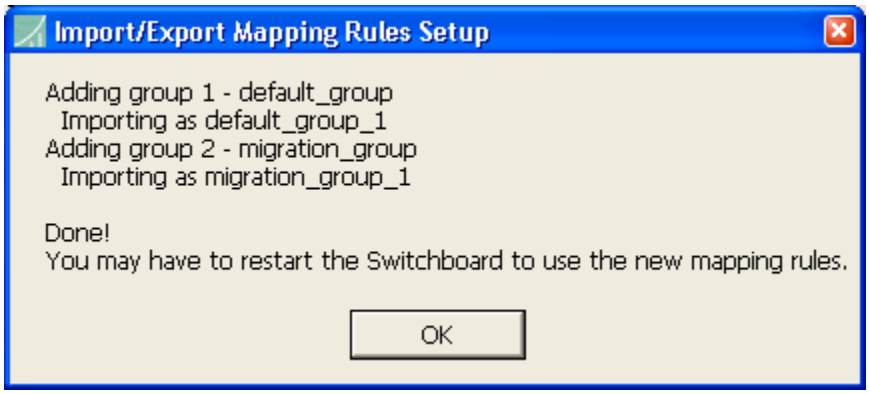


Figure 63 – The results of importing new mapping rules

Note that the DICOM Switchboard will have to be restarted before you can use the newly imported groups of mapping rules. Once you have restarted the DICOM Switchboard, you can use the Mapping Rule Editor to activate the new sets of mapping rules and to edit them, if necessary.

For Linux: From a Switchboard Linux Command Terminal, run the command

```
perl import_mapping_rules.pl
```

and follow the prompts. (See **Figure 64** below.)

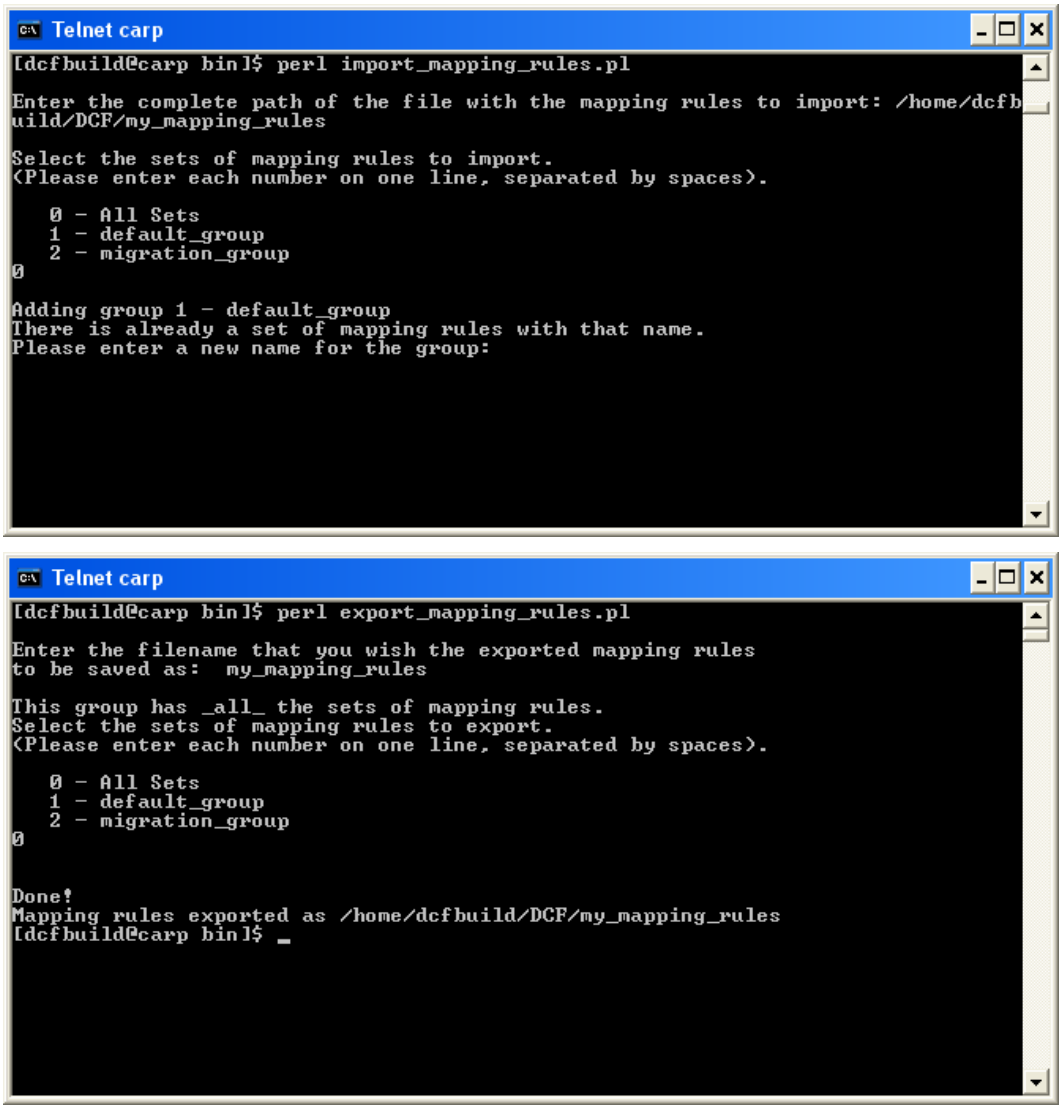


Figure 64 – Importing and exporting mapping rules on Linux

13 License and Activation

13.1 Install new license

If your DICOM Switchboard license has expired, you may request a new one from Laurel Bridge Software. This utility, which is available from the Windows Start Menu, is used to install your new license for you. (For this utility, the abbreviation DSB = "DICOM Switchboard".) Note that in some situations, you may need to restart the Switchboard service or reboot your computer before the new license will take effect.

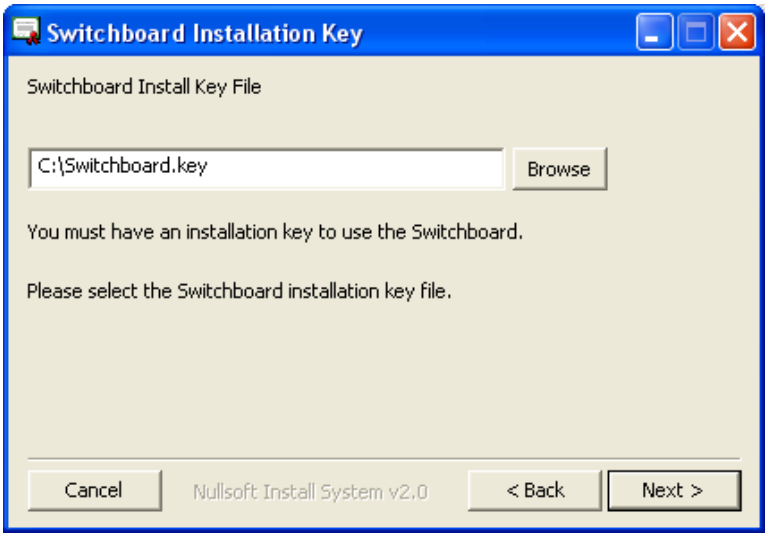


Figure 65 – Installing a new license

For Linux: Install a new license by copying the key file as follows:

```
cp <keyfile> cfg/systeminfo
```

13.2 Activating a license

When you install some licenses for the Switchboard, it works right away; other licenses may require activation before the Switchboard will work – this is especially true if you were given just a Product Serial Number for activation. If you need to activate your license, you may see a warning message when you first install the Switchboard, or you may see a message like this at the bottom of the Switchboard's Operations window.

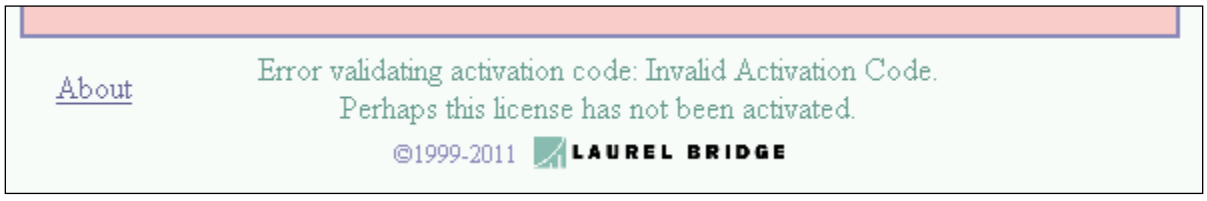


Figure 66 – Warning to activate the license

To activate your license, launch the License Activation Utility from the Start menu:

```
Start → All Programs → Laurel Bridge Software → Switchboard →
Utilities → Activate license
```


The License Activation Utility will let you activate your license in either Network mode or in Manual mode; each is described below. Note that some keys do not require activation – in this case, the utility will warn you to this fact and you can only exit the tool. (Note that due to UAC restrictions, you may have to launch the utility with administrative privileges – right-click on the Start menu shortcut and click "*Run as administrator*".)

13.2.1 Network Activation

If you have Internet connectivity, you will want to activate the license via the Network – you will see a screen like that shown below.

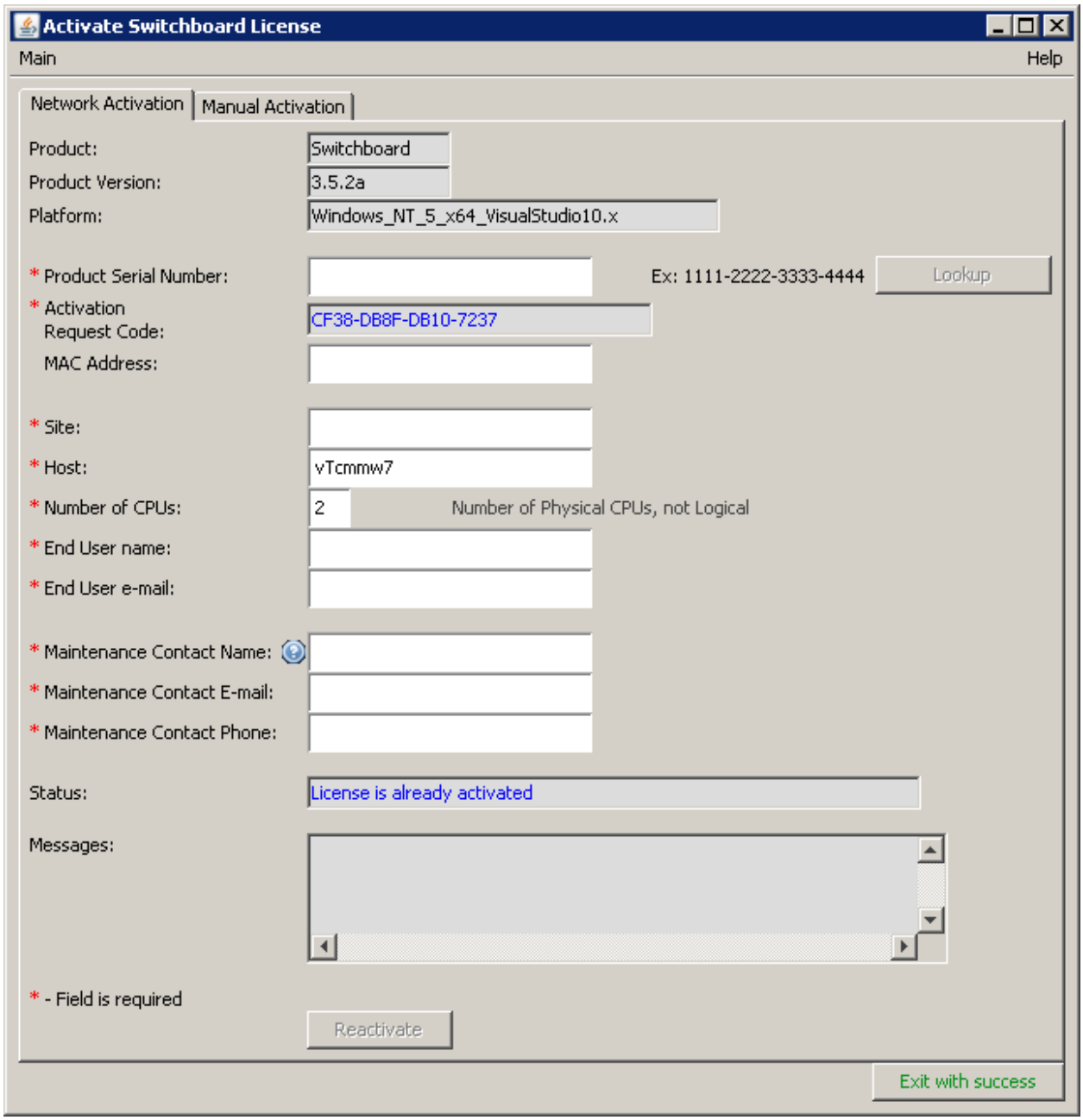


Figure 67 – Network mode for activating a license

Fill in all of the fields – only the MAC Address is optional. The Product Serial Number was given to you when you purchased the Switchboard, or it can be found on the LBS licensing web site as you view your keys. (Once you have entered the Product Serial Number, you can use the Lookup button to query the Laurel Bridge Software website for any existing data for the key.) The Maintenance Contact is the person who Laurel Bridge Software should contact at **your company** when the application is due for renewal of its software maintenance contract; it is **not** tech support. Note that the fields in blue do not need to be entered by you – the Activation Request Code is a system identifier that is generated on your computer by the Switchboard.

Once all the fields are filled in correctly, press the Activate button. The utility will communicate with the Laurel Bridge Software licensing web site and receive an Activation Code and other information back from the web site. Upon success, the status fields will look something like this:

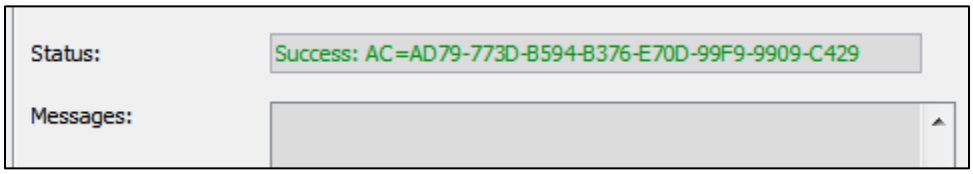


Figure 68 – Activation succeeded via Internet

The Switchboard license should now be activated, allowing the Switchboard to be fully functional – note that you may need to reboot the computer if you are activating the license for the first time. If activation failed, you will see error messages explaining why. Resolve the errors if possible and try activating again.

Note that in some circumstances, you may need to register the Switchboard's System Manager in order for the Switchboard to run – open a Switchboard Command Prompt and run "`perl bin/RegisterSysMgr.pl`". This may be required, for example, if you did not activate the license when you first installed the Switchboard.

13.2.2 Manual Activation

Manual activation is used when the computer with the Switchboard does not have access to the Internet and to the Laurel Bridge licensing web site – note that Network Activation is the *preferred* mode.

After you launch the License Activation Utility, you should select the Manual tab if it is not already selected.

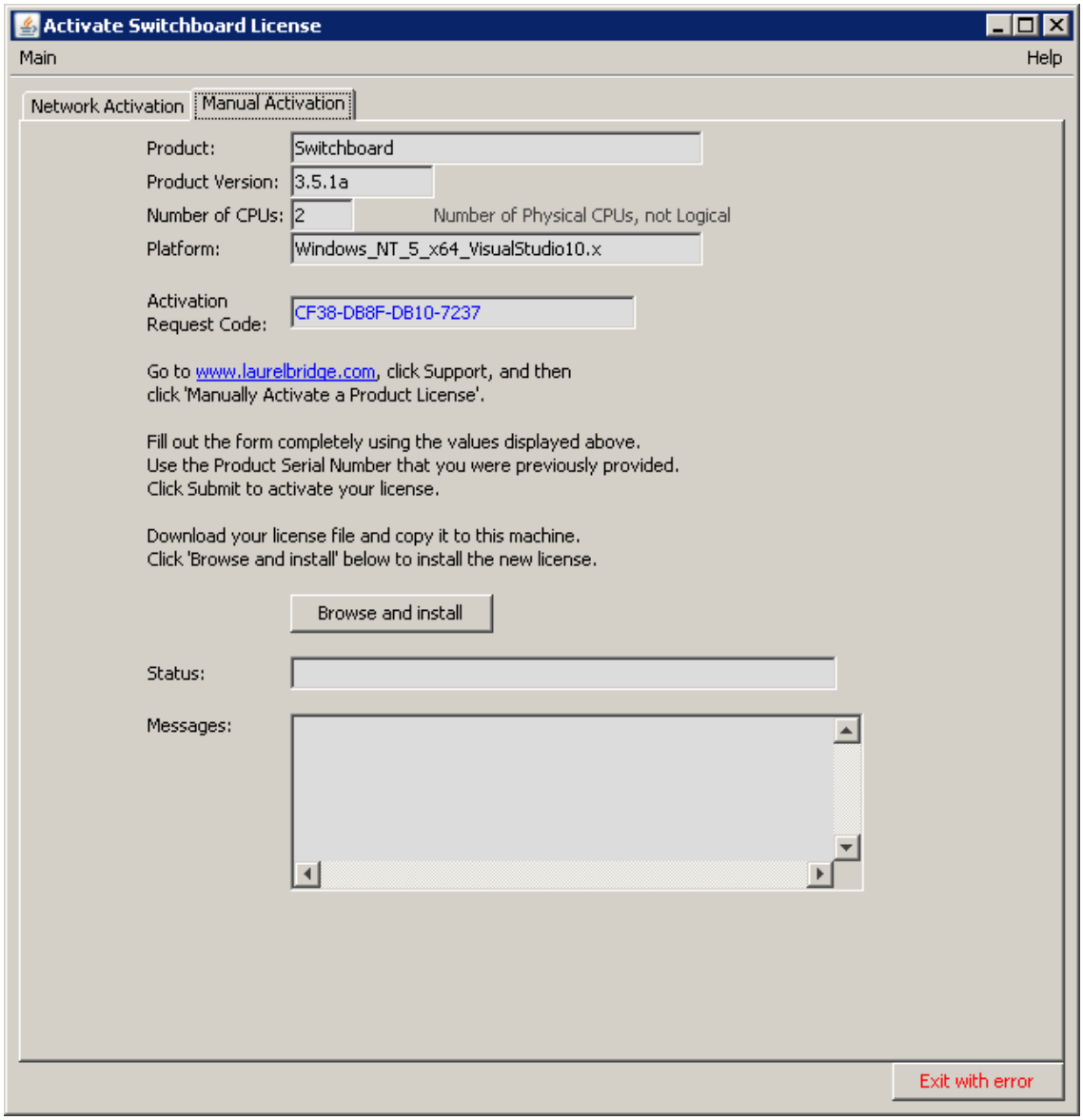


Figure 69 – Manual mode for activating a license

Using a web browser on a different system, proceed to the Laurel Bridge Software customer web site, select "Support", and then select "Manually Activate a Product License" (or click this link: https://www.laurelbridge.com/product_activation.php). Enter the Product Serial Number that was obtained and then the Activation Request Code displayed by the utility (in the example above, it is CF38-DB8F-DB10-7237). Choose the version of software that you are activating. Also enter the site and contact information, and the number of CPUs for the system that is being activated. See the following screenshot:

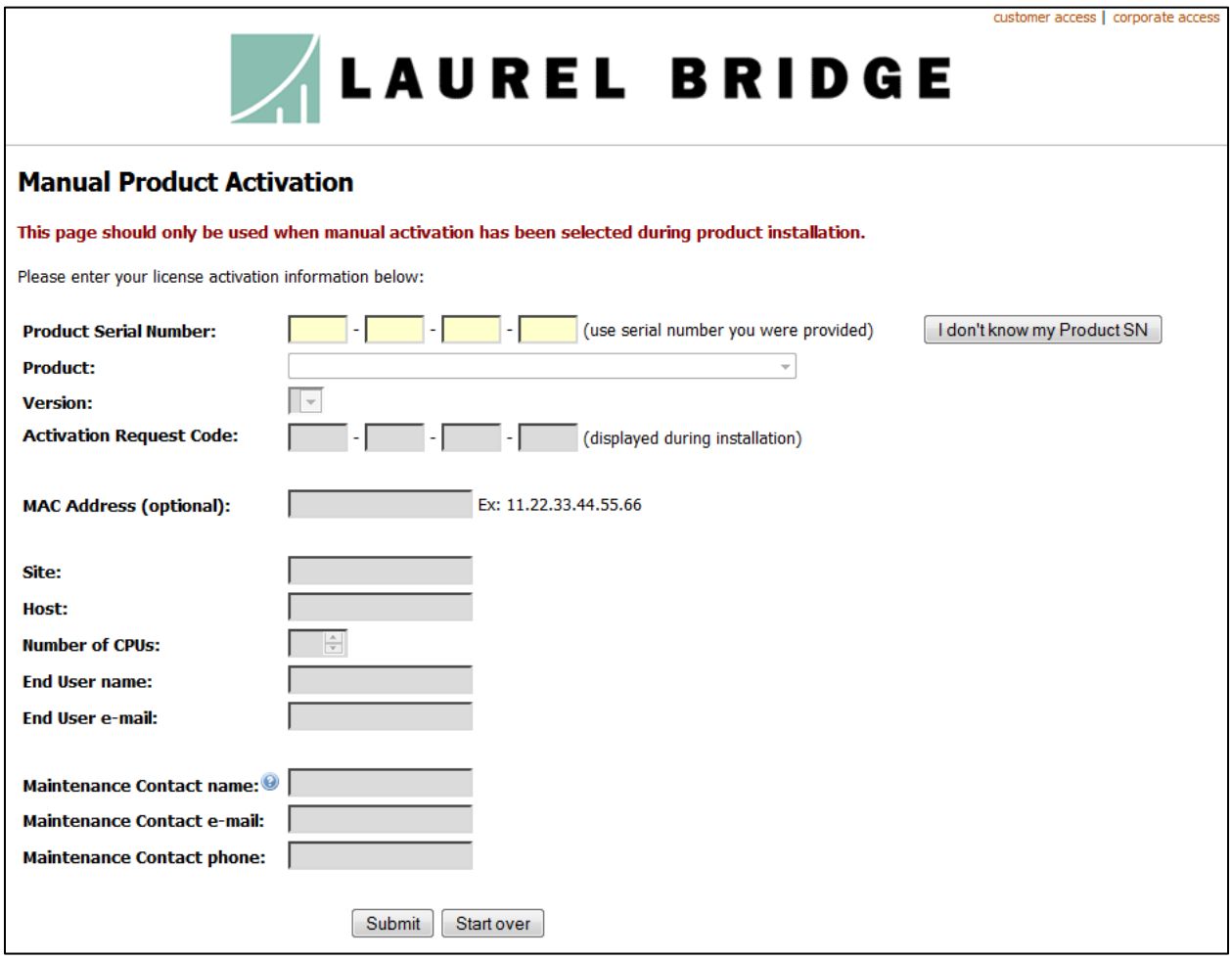


Figure 70 – License activation web page

After you click Submit, you will see a screen like that below.

Figure 71 – Web page showing license activation code

Click the Download button and save the license file, and copy it to the target machine. Then click "Browse and Install" on the License Activation Utility – search for the license file and select it. The utility will install it and verify that the license is valid. When it is done, the utility should look similar to this:

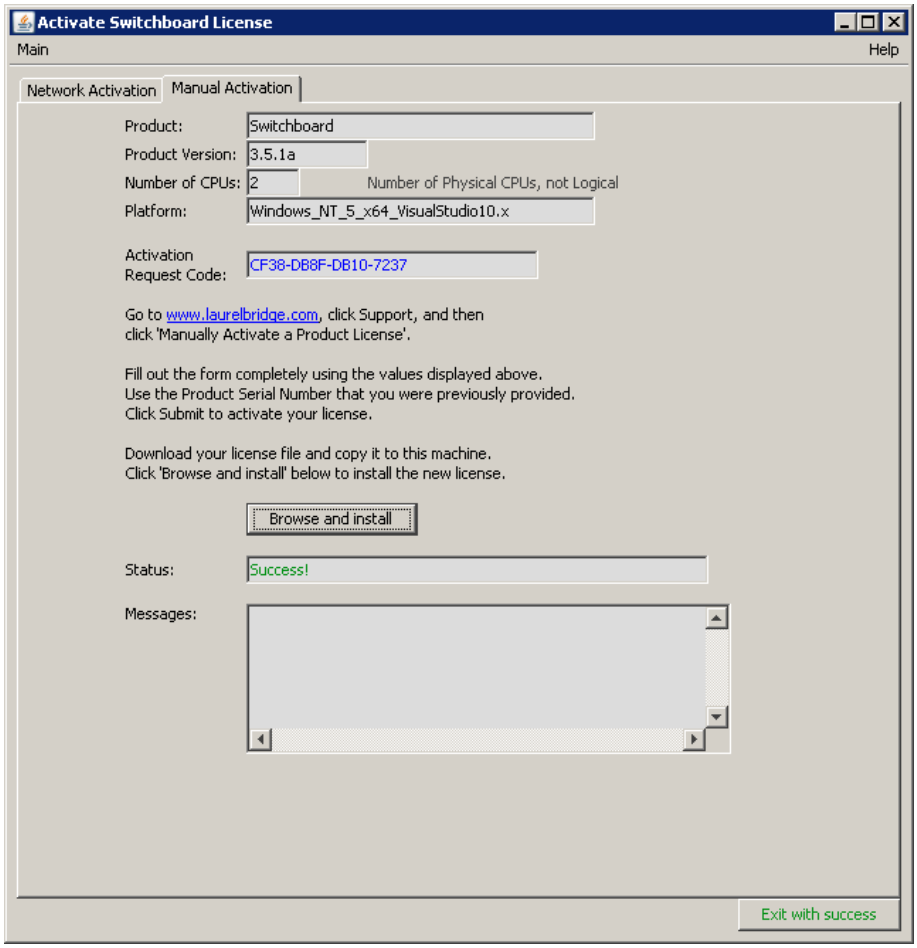


Figure 72 – Successfully activated the license manually

The Switchboard license should now be activated, allowing the Switchboard to be fully functional. Note that you may need to reboot the computer if you are activating the license for the first time.

Note that in some circumstances, you may need to register the Switchboard's System Manager in order for the Switchboard to run – open a Switchboard Command Prompt and run "`perl bin/RegisterSysMgr.pl`" (you may need to open the command prompt with "Run as Administrator" if UAC restrictions are in place). This may be required, for example, if you did not activate the license when you first installed the Switchboard.

14 DICOM Switchboard Applet Issues

Note: As of version 3.3.40c, the Switchboard uses web application versions of the Status Monitor and other applets that are more browser-friendly. However, the old Java versions are still available and may be used in a non-applet mode. In addition, some of these applets may still be accessible via the "DCF: Advanced Options" page. However, it is recommended to use the newer web applications and not the applets. The information on applets in this section is deprecated.

Java applets are used to provide part of the functionality of the DICOM Switchboard application. They are used for various tasks, including editing the mapping rules and viewing the log output in real-time. These applets provide a simple interface that can be accessed from anywhere that has web access to the system where the DICOM Switchboard is installed and running. While you will typically configure the DICOM Switchboard while sitting at the computer where it is running, you don't *have* to: you can configure and control the DICOM Switchboard from any computer with a web browser that has network access to the Switchboard's host.

In order to use the applets, you ***must*** have the Java plug-in installed on your system, as well as in the Web browser you are using. (This step is normally done as part of the initial install and setup of the DICOM Switchboard.) The plug-in is needed because some advanced Java components are used that may not be part of the default Java VM in some browsers, especially older ones. The Java plug-in is designed to provide consistency across web browsers. It can be downloaded from the Java site, at **http://www.java.com** as part of the Java Runtime Environment (JRE). Since it is installed as part of the JRE, it may already be installed if you have that on your system. If you do not have the plug-in installed, when you try to start an applet all you will probably see is an unchanging gray box with the text "`Loading Java applet`". In this case, please download and install the plug-in; you may need to restart your browser for this change to take effect.

Some firewalls may warn when a web browser tries to run one of the Switchboard's applets; they may even block the applet from running. If this happens, you should register your browser as an exception with the firewall to unblock access to the browser/applet and to allow it to function normally.

The Java applets used by the DICOM Switchboard use a signed JAR file. This is due to a known bug in Java's CORBA implementation for applets, which causes a security exception to be thrown. (See the Bug IDs in Sun's bugs database: 6203567, 5031209) At this time, when a user downloads or runs a DICOM Switchboard applet in their web browser, the user will be prompted to accept the signed applet from "laurelbridge.com" (see the example figure **below**).

Some applets in the DICOM Switchboard may cause you to be prompted *twice* to accept the signed applet – this is because the applets are split up into smaller JAR files for efficiency; this is not an error. Most browsers will let users view the certificate for the JAR file before accepting it. If the users select "always", they won't be prompted again about accepting applets in that JAR file. Some users may be concerned that accepting the applets will open up security holes on their system – this is not the case, as the applets make no changes to the local box.

Note: If you are using a Firefox web browser, as of version 3.6, you must use the Next Generation Java Plug-in. This comes with JRE 6 Update 10 and newer and should be automatically installed into your browser when the JRE is installed. (Java may be downloaded from http://www.java.com.)

Figure 73 – Example security warning from Firefox for a signed JAR file

Figure 74 – Example security warning from Internet Explorer for a signed JAR file

Note that the Switchboard does provide scripts on Linux (.bash) and Windows (.bat) that can be used to launch most of the executables in non-applet mode; these are noted in the sections about each of the editors and applets. Non-applet mode can be useful, for example, if you are running a browser that does not support the Java plug-in.

14.1 Other Applet Issues

If the applets are not working for you, or maybe they work for you when connecting on one computer but not from another, there are many possible problems – Java, firewall settings, plug-in issues, etc.

14.1.1 Enabling Java

If the applets aren't working for you, one possibility is that Java is disabled in your web browser. You should re-enable it (you may have to restart your browser) and try them again.

On Firefox, the panel for controlling Java looks like this:

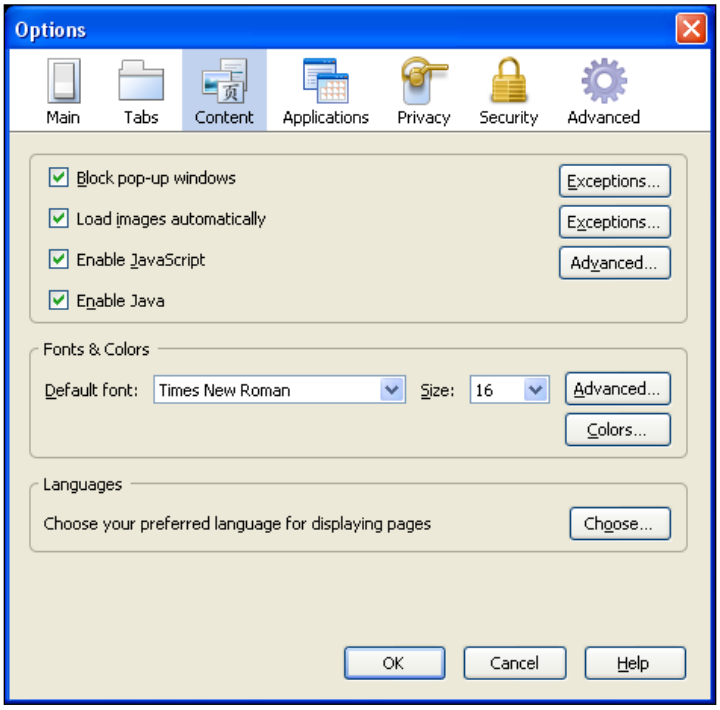


Figure 75 – Enabling Java in Firefox

On Internet Explorer, the panel to enable Java looks like this:

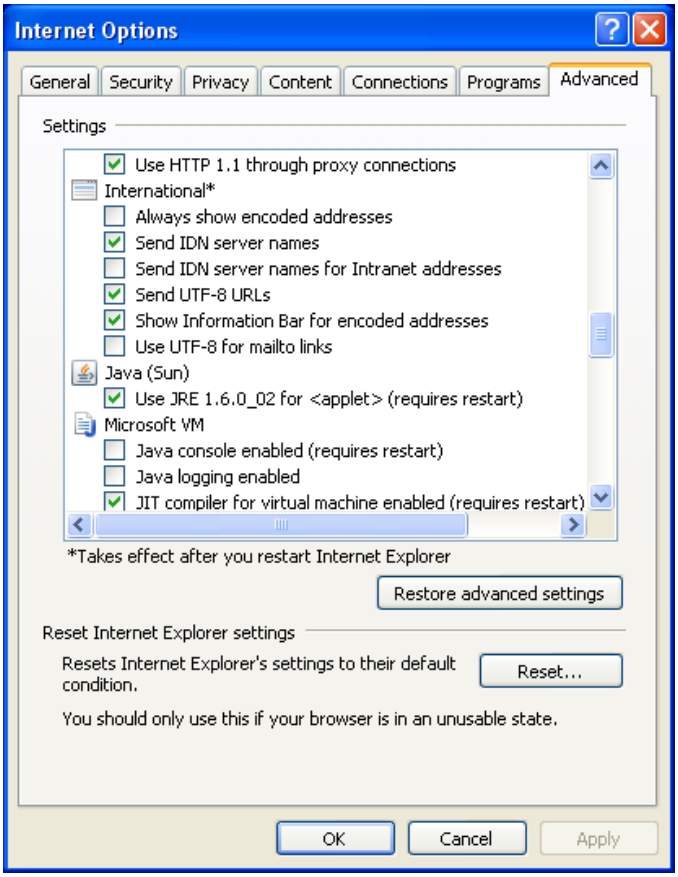


Figure 76 – Enabling Java in Internet Explorer

A similar issue could be that you have an older version of the Java plug-in, one that does not have the correct components for running the Switchboard's applets. The DICOM Switchboard is designed to run with Java 6 Update 2 (formerly called "1.6.0_02") or newer (as seen in the **above** screenshot). If you have an older version, you should download a newer version from "http://www.java.com" and install it.

14.1.2 Firewall Issues

Another possibility if the applets are not working for you – especially if you are trying to connect from a remote box – is that your firewall is configured in such a way that the applets cannot run correctly. (It is even possible that the firewall may block all communication between the remote applets and the Switchboard.) This could include that the firewall is set up to block Java from downloading, or that it needs to be configured to allow the Switchboard's applications (including Java) to run and to communicate, or they could be blocking communications between the applets and the Switchboard's services. While you should not disable your firewall, you should configure it to allow the Switchboard and its applications to communicate by setting them as exceptions. For example, **below** is a screenshot of what this looks like for the default Windows Firewall. (Note that the example below shows that the dcf_switch has been enabled as an exception but that Java has not *yet* been enabled.)

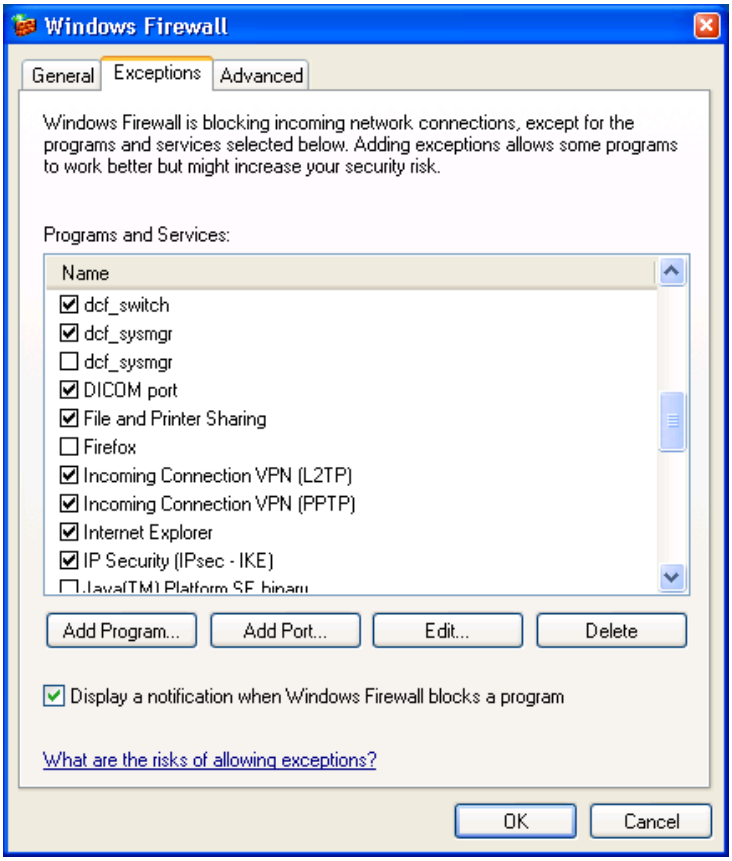


Figure 77 – Setting firewall exceptions

14.1.3 Other Firewall-related Issues

Another problem that may occur is if you are using the web browser on Machine A to control the Switchboard on Machine B, and you find that the applets are not working. In such a case, one possibility is that the firewall (or possibly a proxy server, if you are using one) is preventing the applets from connecting to the Switchboard's configuration and logging servers.

The Switchboard uses CORBA servers to handle logging and configuration management. When these CORBA servers start, they may be listening to a different port each time they come up. The published "stringified interoperable object reference" or "oref" file describes the port and host that a given server is using. A client with that info can then always connect. With the firewalled situation, that doesn't work so well.

You don't want to open up every port in your firewall just in case the Switchboard needs that port to communicate – that is too dangerous. What you can do is instruct the servers to use a fixed port – the applets on the browser system will then try to connect to that fixed port. Then you can modify your firewall to open only those ports for that use. (You do need first to ensure that something else isn't going to want that port.)

To do this, manually edit the configuration file %DCF_CFG%\systems\dsb_switch_win32.cfg (on Linux, this would be $DCF_CFG/systems/dsb_switch_unix.cfg.) Find the sections in the configuration file for the Distributed Configuration Data Service (DCDS_Server) and the Distributed Logging Service (DLOG_Server); you may also need to modify the section for the Real-Time Log Server (RTLog_Server). Modify the startup comments in each section to provide a Java property to force the port for the server to always be the same. (In the example below, we used 7777 for DCDS, 8888 for DLOG, and 9999 for RTLog – see the string "`-Dcom.sun.CORBA.ORBServerPort=7777`".)

Run "netstat –a" to see what ports are in use, and select three (preferably higher than 2000, say) and edit these commands in the system startup file accordingly. Those would be the ones to open in the firewall.

```
 [ DCDS_Server.001 ]
type = server
checkpoint_file = $DCF_VAR{DCF_TMP}/dcds_server.oref
command = java.exe -Dcom.sun.CORBA.ORBServerHost=myhostname
    -Dcom.sun.CORBA.ORBServerPort=7777 com.lbs.DCDS_Server.DCDS_Server
    -appcfg /apps/defaults/DCDS_Server -appicfg /procs/DCDS_Server.001
post_start_delay_seconds = 0
shutdown_timeout_seconds = 3

[ DLOG_Server.001 ]
type = server
checkpoint_file = $DCF_VAR{DCF_TMP}/dlog_server.oref
command = java.exe -Dcom.sun.CORBA.ORBServerHost=myhostname
    -Dcom.sun.CORBA.ORBServerPort=8888 com.lbs.DLOG_Server.DLOG_Server
    -appcfg /apps/defaults/DLOG_Server -CDS_a_use_fsys
    -appicfg /procs/DLOG_Server.001
post_start_delay_seconds = 0
register_app_timeout_seconds = 60
shutdown_timeout_seconds = 3

[ RTLog_Server.001 ]
type = server
command = java.exe -Dcom.sun.CORBA.ORBServerHost=myhostname
    -Dcom.sun.CORBA.ORBServerPort=9999 com.lbs.RTLog_Server.RTLog_Server -
appcfg /apps/defaults/RTLog_Server -appicfg /procs/RTLog_Server.001
post_start_delay_seconds = 0
shutdown_timeout_seconds = 3
```

Save the changes to the configuration file and then restart the Switchboard.

14.1.4 Hostname issues

If the Switchboard won't start or your applets won't work, another possible reason is that the hostname for your computer does not resolve to a valid IP address. For example, this may occur if you are using the Switchboard on a machine whose name is not known by your DNS server. In such cases, the "oref" file – which describes the host and port that the servers should connect to – has incorrect information: a connection to that *host* will fail since the hostname does not map to a valid IP address. (See the previous section for a more complete explanation of the CORBA servers and the oref files.) (You can see what host the oref file is looking for by going to this url – http://www2.parc.com/istl/projects/ILU/parseIOR/ – and pasting the contents of the oref file into its input field. If the host referenced in the output is not the same as your hostname, you may have a DNS problem.)

To fix this, edit the startup configuration file `%DCF_CFG%\systems\dsb_switch_win32.cfg` (on Linux, this would be $DCF_CFG/systems/dsb_switch_unix.cfg.) Find the sections in the configuration file for the Distributed Configuration Data Service (DCDS_Server), the Distributed Logging Service (DLOG_Server), and the Real-Time Log Server (RTLog_Server). Modify the startup comments in each section to provide a Java property to use a specified hostname – change the argument to ORBServerHost to be a different hostname for the computer that does resolve correctly; you can also specify the IP address or just "localhost". (In the example above, see the string "`-Dcom.sun.CORBA.ORBServerHost=myhostname`".)

Save the changes to the configuration file and then restart the Switchboard.

14.2 JavaScript Issues

JavaScript must be enabled to use the DICOM Switchboard and its web pages properly. If JavaScript is not enabled, you may see error messages like those shown **below**:

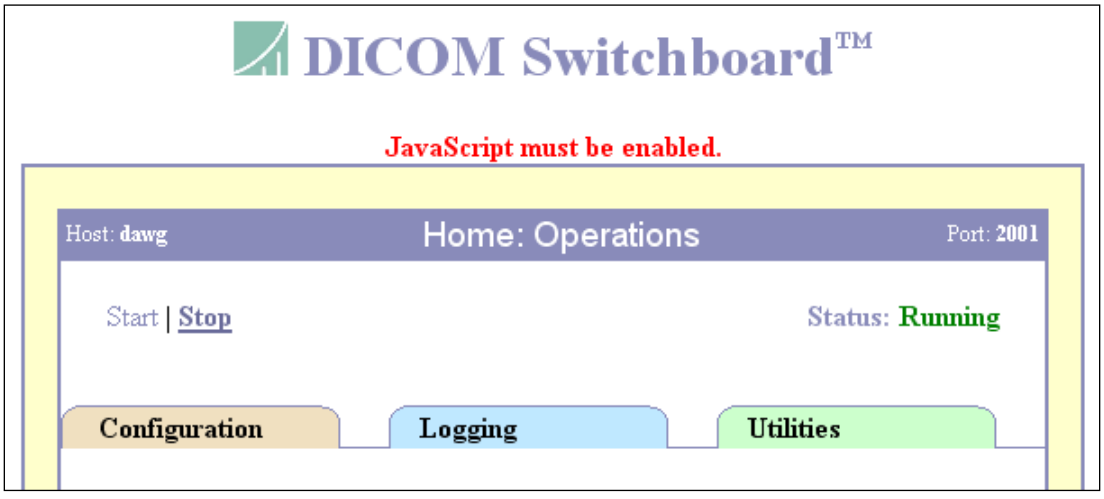


Figure 78 – Example JavaScript warning from Firefox

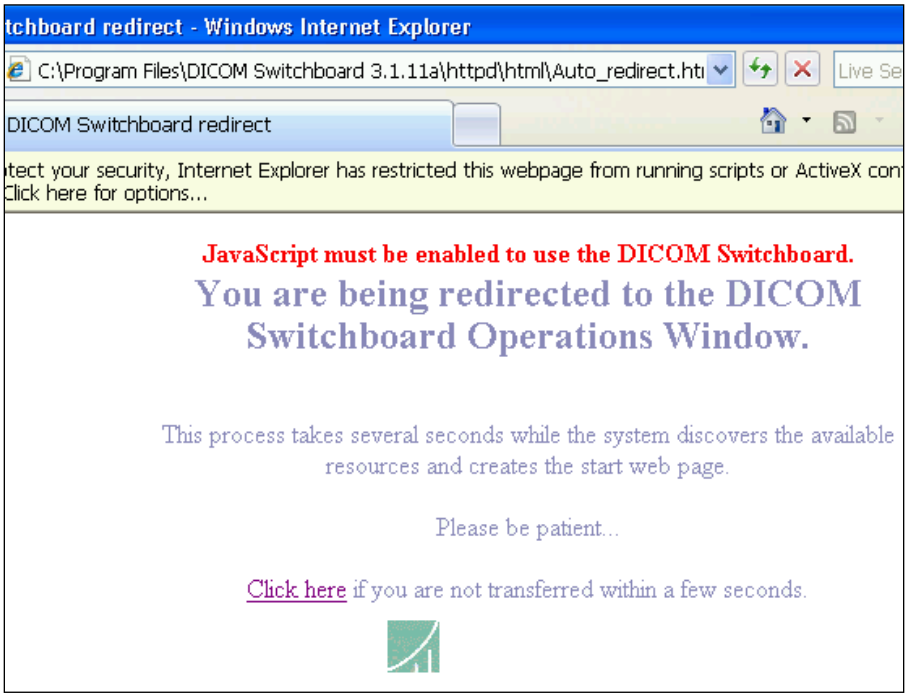


Figure 79 – Example JavaScript warning from Internet Explorer

Each web browser has its own way of enabling and disabling JavaScript, so consult the documentation and their user manuals for specific information on how to enable JavaScript.

For Firefox, select the “Tools” menu, then the “Options” sub-menu. Select the “Content” tab and click “Enable JavaScript”.

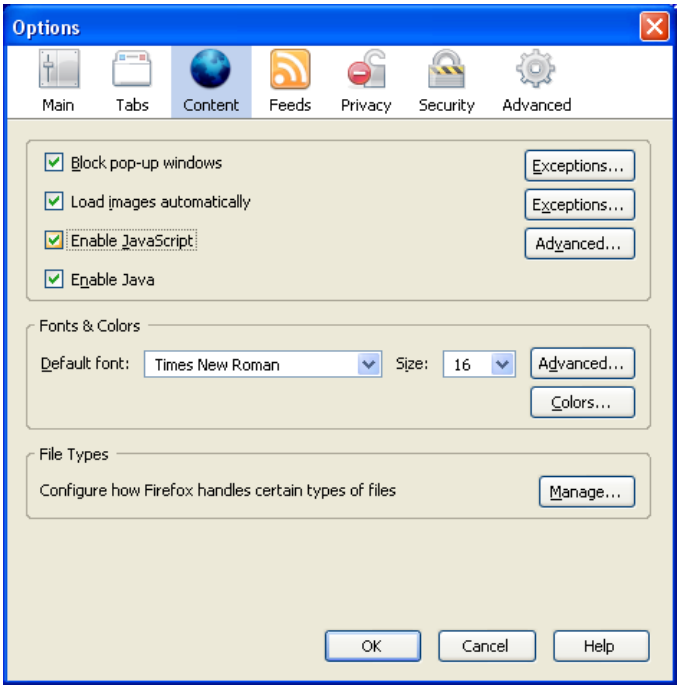


Figure 80 – Enabling JavaScript in Firefox

For Internet Explorer 7+, select the “Tools” menu and then the “Internet Options” sub-menu. Select the “Security” tab, and then select the “Local intranet” zone. Click “Custom level…”; find the “Scripting” section near the bottom, and click “Enable” for “Active scripting”.

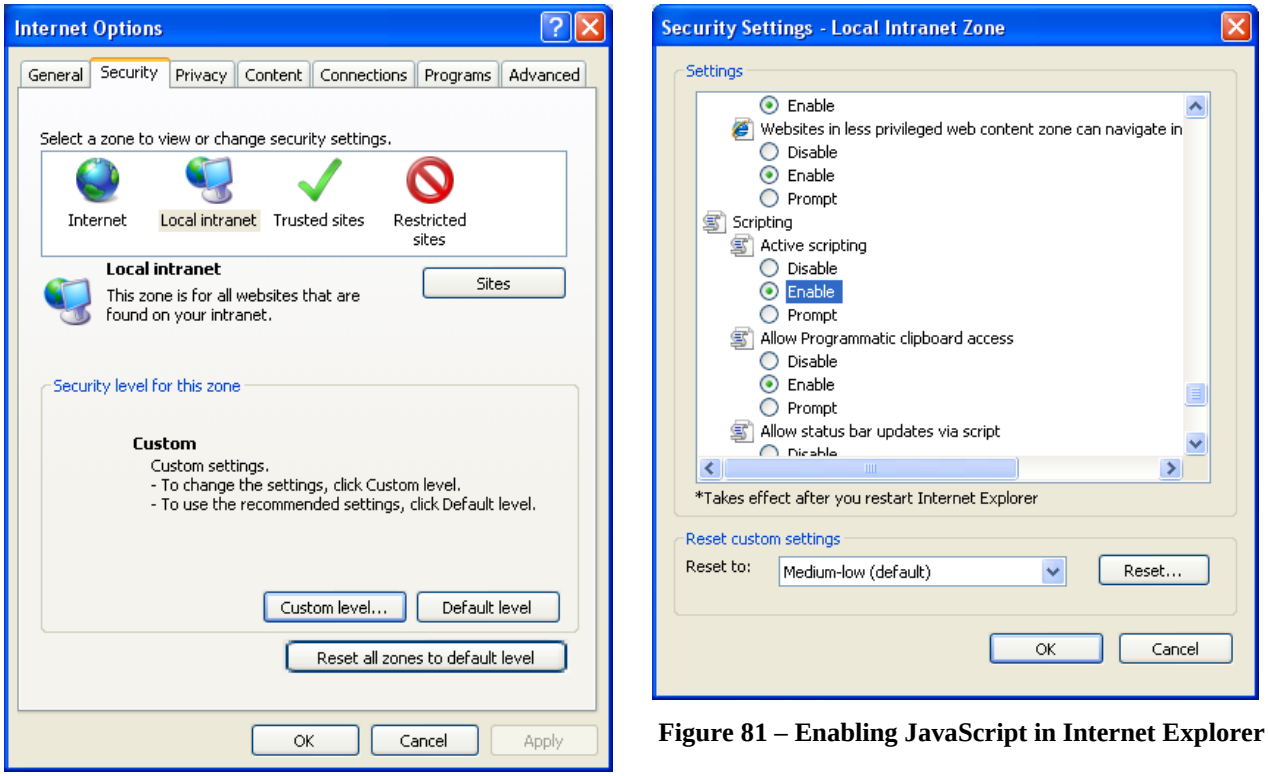


Figure 81 – Enabling JavaScript in Internet Explorer

Alternatively, you can select the “Trusted sites” zone. Click the “Sites button”, and then click “Add” to add http://<your computer name> to the list of trusted sites. You may wish to check that “Active scripting” is enabled for the “Trusted sites” zone, following the steps described above for the “Local intranet” zone.

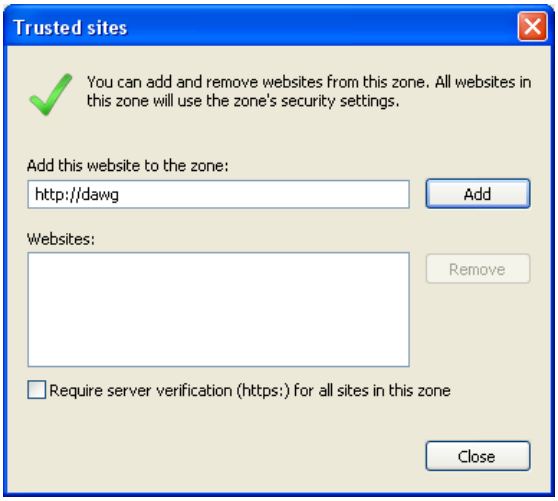


Figure 82 – Adding your computer to the list of trusted sites

Be sure to click “OK” as you close each menu to accept the changes in security preferences.

15 DICOM Switchboard Web Server

The DICOM Switchboard uses web pages and an Apache web server to provide its user interface. Many of the Switchboard's functions can be accessed via the web pages and CGI scripts they use, while other functions use JavaScript-based web applications to provide greater functionality. Using web pages and scripts allows the DICOM Switchboard to be configured and controlled from any computer via a standard web browser such as Internet Explorer or Mozilla Firefox.

However, once you have configured the Switchboard's mapping rules and filters, it is possible that you may wish to "turn off" the web interface; for example, to prevent other users from modifying the configuration. This can be easily done by closing the command window that Apache is running in or pressing CTRL-C in that window.

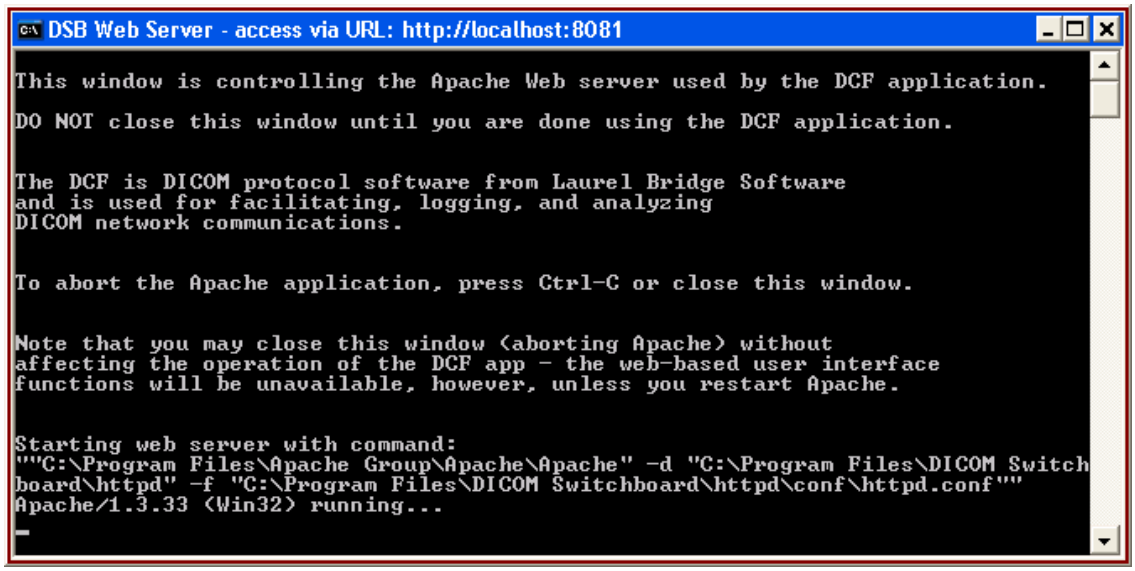


Figure 83 – Command window that is running the Switchboard's web server

The DICOM Switchboard will operate correctly without the web server, although the configuration utilities and other helpful applications, such as the Switchboard Status Monitor (Chapter 10), can be accessed only via the web server. It is also simpler to start and stop the DICOM Switchboard via the web interface than from a Switchboard Command Prompt.

If you have stopped the web server and wish to start it up again in order to access the web-based utilities, you can do so via the Windows *Start* menu:

```
Start → All Programs → Laurel Bridge Software → Switchboard → Start
Switchboard
```

This will restart the Apache web server; doing this will also attempt to restart the DICOM Switchboard, but find that it is already running – this is not an error and will not affect the Switchboard's operation.

(**For Linux**: Open a command terminal to the Switchboard's installation directory, "dot" the dcf.env file to set the environment, and run the command: `run_apache.pl`. **Note** that SUSE 13.2 uses different commands, detailed in Section **3.5.1 Configuring the Switchboard service on Linux** above.)

See **Appendix F: Additional Information on Apache** for information on running the Apache web server as a service and for additional information on configuring Apache with the Switchboard.

15.1 Apache and Firewalls

The DICOM Switchboard is designed to be configured using a web browser talking to a web server – this means that it can be accessed and configured from almost anywhere. But typically, users configure it while sitting at the machine that the Switchboard is installed on.

If you want to access the Switchboard web pages from another machine but find that you are unable to, this is usually because a firewall is blocking access to the Apache web server on the host machine. In such a case, you should modify your firewall settings so that the Apache port can receive communication. You selected the Apache port used when you installed the Switchboard (see Section **3.3 Installing the DICOM Switchboard**), now you need to "open a hole" in your firewall for that port only. (You may have already opened a similar "hole" for communication with the Switchboard application itself – see Chapter **7 Configuring the DICOM Switchboard**.)

See **Figure 77** above for an example of how this might look for Windows Firewall.

16 More Information on Using the DICOM Switchboard

There are many other useful features and options included with the DICOM Switchboard beyond what is described herein. The Switchboard configuration screens and tools have help pages that are accessible from each application's Help menu (much of that information has been included in this User Manual). Application docs are also available from the options on the DCF: Advanced Options page.

If you have questions about using the DICOM Switchboard or need help configuring it, please feel free to contact Laurel Bridge Software at mailto:support@laurelbridge.com.

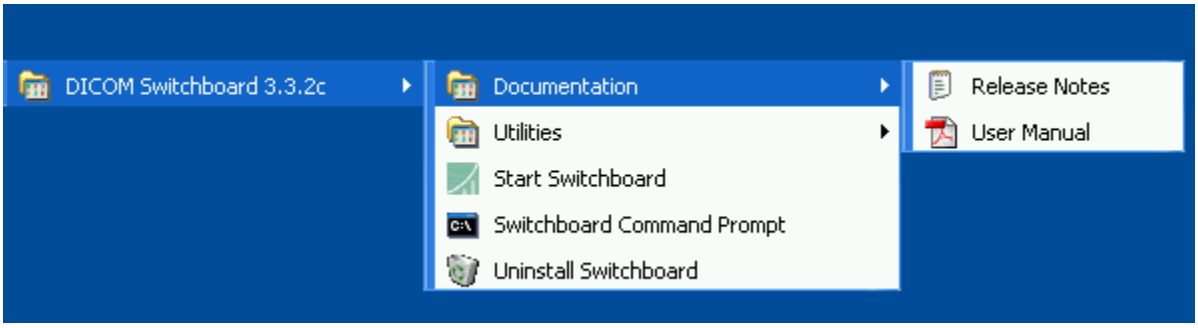


Figure 84 – Accessing Documentation for the DICOM Switchboard

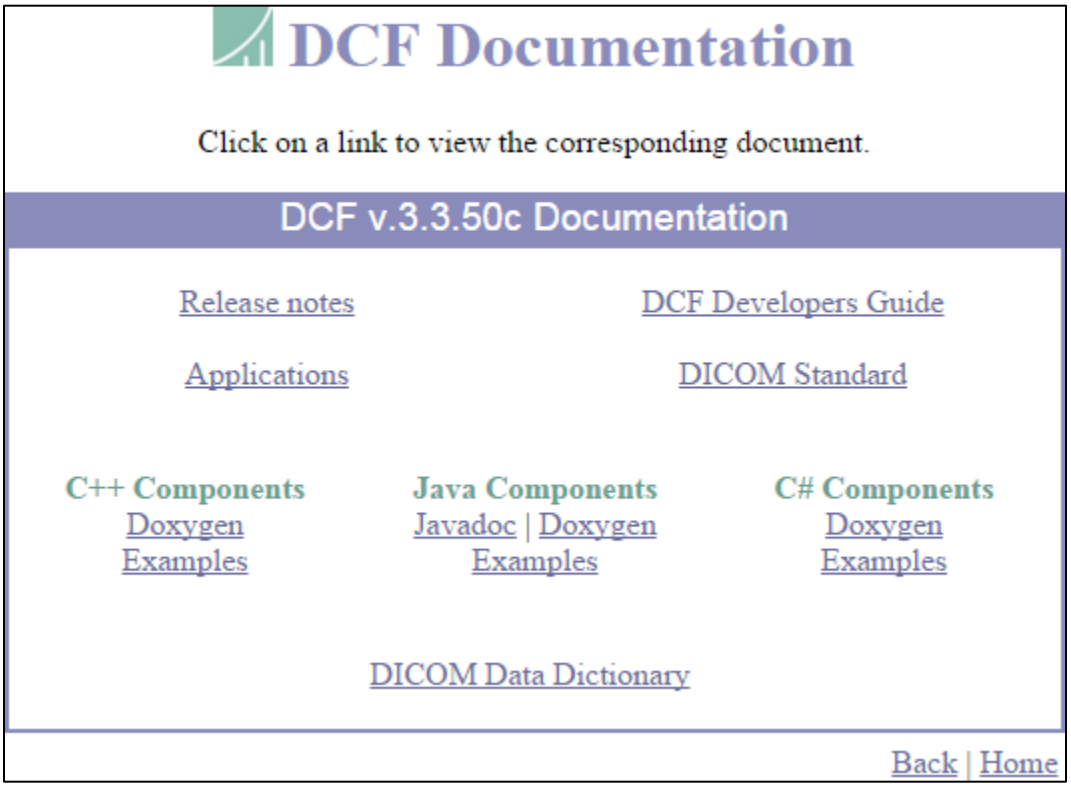


Figure 85 – Additional DCF Documentation

Appendix A: Example Filter: Modifying Elements

This appendix shows the built-in help text related to modifying elements. The information below is accessible from within the DICOM Switchboard by selecting the "Help" option in the edit window for the Modify Filter. (Help is also available within the program for the other filter types.)

The Modify Filter option represents one of the most powerful features of the DICOM Switchboard.

1 Overview

Filters in the DCF may be used to modify the data in a tag. Most other filters will modify which DICOM elements are placed in a DIMSE message; the "Modify Filter" is designed to modify the *values* in those elements, as well as allowing you to move or copy the data into other elements.

For example, suppose your software expects the patient's name to be in your proprietary DICOM tag "abcd,abcd", but you are not getting the patient name data in that tag from the modalities. The "Modify Filter" allows you to specify that the patient name (tag 0010,0010) be copied to your tag.

Or suppose that your software expects the patient name to have the first name first, instead of the DICOM default of last name first. The "Modify Filter" allows you to use regular expressions to switch these elements around. You can even adjust the case of the data values, should that be desirable.

2 How the Filter Works

When the "Modify Filter" is applied to the data in a DICOM tag, the regular expression in the "Old value" field is applied to it. If the regular expression (or "regex") doesn't match, no operations are performed on the tag's data. If it does match, the "New value" field indicates how the matching items from the regex are used to reconfigure the data. The new data may also have its case changed so that all alphabetic characters are in **UPPER CASE**, **lower case**, or their case remains as they were in the original data.

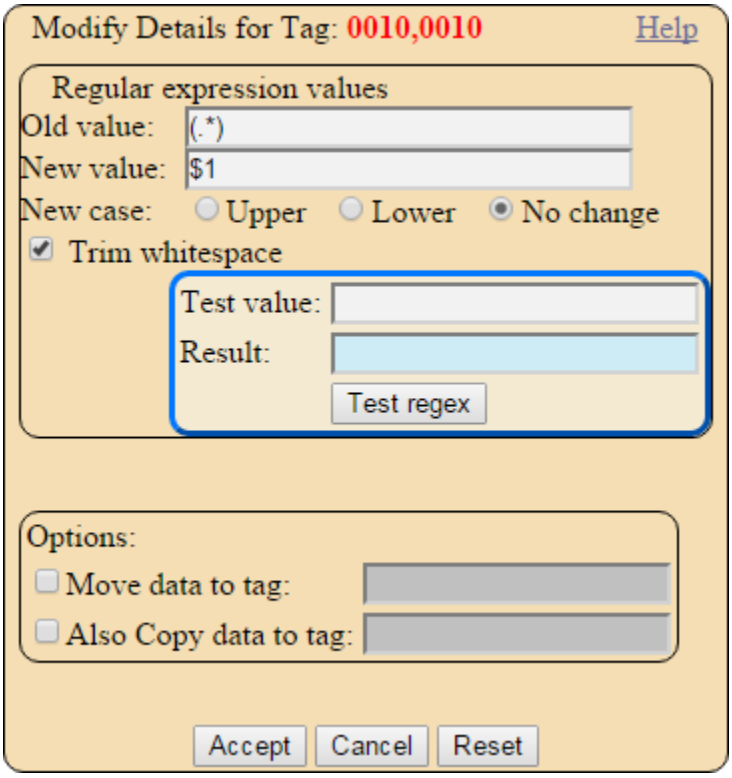


Figure 86 – Creating an Element Modify Filter

Following the reconstruction of the data, it may also be moved to be the value for some other DICOM tag; it may also be copied to another tag. Note that any existing values for those tags will be overwritten with no warning.

- Check the "*Move data to tag:*" box to move the data from the current tag to a new tag, which should be entered into the box to its right.
- Check the "*Also Copy data to tag:*" box to copy the data from the current tag to a new tag, which should be entered into the box to its right. This is useful, for example, if the data is in tag X but you expect it to be in tag Y.

The *modified* value is copied and/or moved to the new tag; if you want to copy the original value to another tag and also modify the value in the current tag, then you will have to create multiple filters, one for each operation.

Note: The values that are copied are the *string* values of the elements by default, since the Modify Filter is designed to work on string type elements. If the Switchboard does not know the VR of a tag, then the element may be treated as a binary element, which could cause a different value to be written than expected. For example, if your dataset is in implicit format and contains a private tag intrinsically of type LO with the value "Bob", then since the Switchboard does not know the VR of the tag, the element is treated as if it were of type UN and the value would be interpreted as the hexadecimal bytes "0x420x6F0x620x20" (the hex value plus a padding space) when stringified. To prevent this, it is necessary to add any private tags you may plan to process to the Switchboard's Extended Data Dictionary. (This procedure is described in more detail in **Appendix E: Editing the Extended Data Dictionary**.) *This should be done for any private element that should not be interpreted as type UN*. Once you are done editing the Extended Data Dictionary, restart the Switchboard and it will be ready to properly process your tag and its value.

3 Modifying Data Using Regular Expressions

The DCF "Modify Filter" uses Perl-style regular expressions (regexes) to match data in strings and to modify them. You may be familiar with Perl or other code that uses regular expressions to match data, such as the use of "`m/ell/`" to match any expression that includes "ell" in it, such as "hello" and "jelly", but not "balloon".

Matching via regular expressions can also be used to extract parts of a string that match by using the **grouping metacharacters ()**. In Perl, the parts that match each set of grouping characters are put into special variables $1, $2, and so on. The "Modify Filter" uses this same syntax to match and extract elements and to allow you to specify how they should be reordered in the resulting "New value".

In the Modify Filter dialog screen, the "Old value" field indicates how the data should be matched and grouped, while "New value" specifies what to do with each grouping.

For example, suppose that you are receiving a patient name that has a space for a separator instead of the standard caret (^). You could specify that the space should be converted to the caret by filling the fields as follows:

- Old value: `([^ ]*) ([^ ]*)`
- New value: `$1^$2`

The Old value expression specifies anything that is not a space, followed by a space, followed by another set of characters that is not a space. Notice that, within the brackets, the caret is used to indicate "*not*". The "*" indicate 0 or more of the preceding thing, so the expression `( *)` means 0 or more spaces and the expression `([^ ]*)` means 0 or more things that are not spaces.

In the New value expression the first set of characters selected `([^ ]*)` goes into $1, and the second set `([^ ]*)` becomes $2. The New value `$1^$2` says to rebuild the data as the first set followed by the second set, with a caret between them (here, the caret is a literal character, not a "not" metacharacter).

Using this regex, the string "john doe" would become the string "john^doe".

3.1 Testing your Regular Expression

Using regular expressions is not always easy, and you may want to see if what you specified produces the desired result. If the DICOM Switchboard has been started and is running, the Modify Filter's user interface provides a mechanism for testing your regular expressions.

Below the fields for entering the regular expression's values, you may enter a value to test. When you click the "*Test regex*" button, the "Result" box will display the results of applying the regex to the test value. Please note that the test value data will not change if the regex doesn't match. A warning message may appear for some failures with the test.

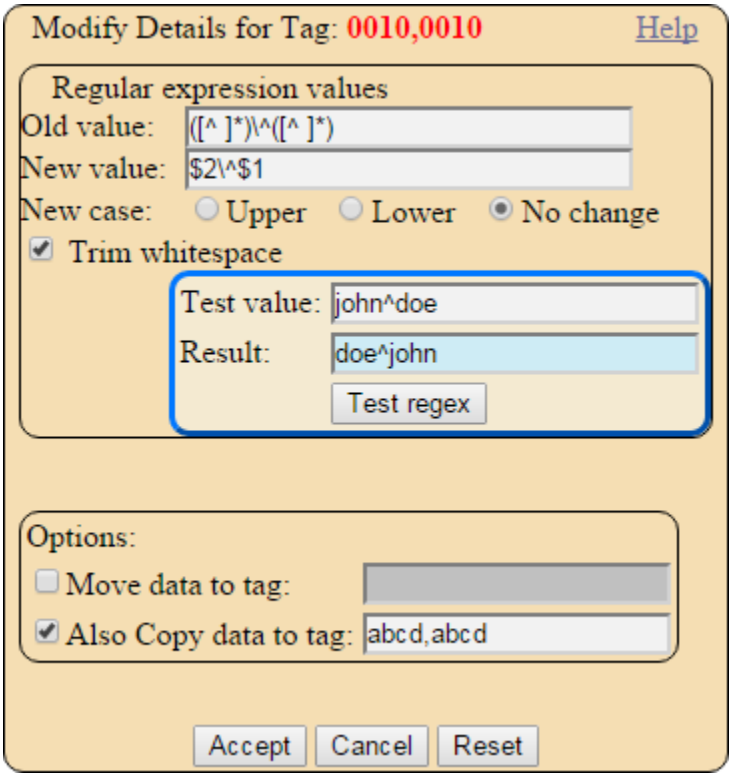


Figure 87 – Testing a Regular Expression

This makes it easy to test your regular expressions until you get the desired results.

3.2 Regex Examples

1. Match the entire string.
   - Old value: `(.*)`
   - New value: `$1`
   - Result: "john doe" becomes "john doe".

   Note that these are the default values used for the Old and New value fields.

2. Apply the old "fortune cookie" interpretation trick of tacking "under the covers" to every string you encounter. ☺
   - Old value: `(.*)`
   - New value: `$1 under the covers`
   - Result: "`john doe`" becomes "`john doe under the covers`".

3. Change a space separating two strings to a caret (^).
   - Old value: `([^ ]*) ([^ ]*)`
   - New value: `$1^$2`
   - Result: "`jane doe`" becomes "`jane^doe`".

4. Swap two strings separated by a caret (^).
   - Old value: `([^ ]*)\^([^ ]*)`
   - New value: `$2\^$1`

- Result: "`john^doe`" becomes "`doe^john`".

Note that the caret separator is specified as a literal caret, not a "not" indicator, by prefixing it with a backslash (\).

5. Substitute all spaces in the string with carets.
   - Old value: `([^ ]*)( )`
   - New value: `$1^`
   - Result: "`a b c d e`" becomes "`a^b^c^d^e`".

3.3 Regular Expression Quick Reference

A brief explanation of a few regular expression symbols is listed below.

Metacharacters: **{ } [ ] ^ $. | * + ? **
Note that the special metacharacters can be matched in a regex by putting a backslash before them.

Character	Meaning	Character	Meaning
^	Not	|	Or
*	0 or more times	+	1 or more times
.	Matches any character	$	Put at the end; the match must be found at the end of the string
( )	Groupings	[]	Character classes

Character classes: Character classes are designated with square brackets `[]`; the set of characters to match are placed inside the brackets. Dashes (-) may be used to indicate ranges, such as `0-9`. `[abcde]` and `[a-e]` are both expressions that will match letters "a" through "e".

Some special character classes can be abbreviated:

- `\d` is a digit and represents `[0-9]`.
- `\s` is a whitespace character, such as space, tab, and newline.
- `\w` is a word character (alpha-numeric character), `[0-9a-zA-Z_]`.

Groupings: Use parentheses to group characters together for extraction. See the other examples for illustrations of how grouping works.

3.4 Regular Expression Reference Sources

A complete explanation of regular expressions is beyond the scope of this document. Many references are available on the web; a few are listed below.

Please note that the DICOM Switchboard's filters uses Perl *style* regular expressions, but not the complete Perl language syntax.

- http://www.regular-expressions.info/reference.html - Regular expression basic syntax reference

- http://perldoc.perl.org/perlre.html- Perl regular expressions
- http://perldoc.perl.org/perlrequick.html - Perl regular expressions quick start guide
- http://perldoc.perl.org/perlretut.html - Perl regular expressions tutorial

4 Entering Data

When entering data into a tag field within the DICOM Switchboard, enter two values as hexadecimal (base-16) pairs of numbers, separated by a comma in the traditional "`group,element`" format; for example, enter `0020,0010` for Study ID or `0008,002A` for Acquisition Datetime. You must press "Enter" after filling the data in the field. If data is found to be invalid, a warning will be displayed. (See Section **4.1, Entering Sequences**, below for information on entering a sequence as the tag.)

Once a tag has been entered, the values (if required for that filtering option) should be entered into the "Value" column of the appropriate tables. When entering data into a **Tag-Value** table, you should enter the **Tag** data before the **Value** data.

To delete a tag (or tag-value pair) from a list, select it with the mouse, and press the small red "x" next to it in the table.

After you are done editing the data in your filter form, click on the "*Accept*" button to accept the changes and save them to the filter. Press the "*Cancel*" button if you wish to discard the changes you have made. Press "*Reset*" to discard any changes you have made and reset the values displayed in the form to its original state-this is useful if you have made some mistake editing and want to start over with the original values that were in the filter.

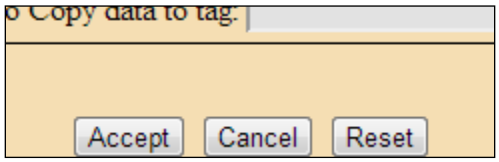


Figure 88 – Saving a Modify Filter

4.1 Entering 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 with sequence - `0080,0100.0008,0060`
- Tag with sequence ID and sequence - `0080,0100.#0.0008,0060`
- Tag with multiple sequences and IDs - `0080,0100.#1.0080,0100.#0.0008,0060`

If no item number is specified, the first item (#0) is assumed. You can also specify the last element in a sequence by "#L" (upper-case is important!) if you don't know how many items are in a

sequence. If you are creating new elements, you can 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 Advanced Applications of the Modify Filter

The DICOM Switchboard allows you to modify data with regular expressions, but what if you want to *match* data via a regular expression? For example, suppose you want to apply your filter only if a certain tag contains a specific string anywhere in its value, such as "`^.*Bob.*$`"? This would be applied if "Bob" occurs anywhere in the string. Or what if you wanted to filter only if "Bob" is at the beginning of the string? The DICOM Switchboard does not (yet) allow regular expressions in the Elements to Match. But with a little creativity, it is possible to mimic this behavior…

Let us suppose that you want to change the value of the Study Description (0008,1030) to something else, say "Laurel Bridge Test Study", if the Accession Number (0008,0050) has a certain value at its beginning; for this example, let's say that value is "LBS". So the data would be filtered/changed if the Accession Number is, for example, "LBS0001" or "LBS0013", but not when the accession number is "DZY0042".

It is not possible to directly use a Modify Filter, which does have regex capabilities, to match the data, since the data will be modified by the regex. Instead, first we copy the data to another tag – a temporary, private one that is not used by anybody else – and then we modify that tag and its data, and compare it to decide if the change should be applied; then we clean up. (See the **Note** on private tags at the end of this section.)

1. Create a new Modify Filter. Set the Elements to Modify to be 0008,0050. In the details for the filter, select "*Also Copy Data*" and enter the temporary tag (say, 0019,0017).

   This copies the value to a temporary tag so that we can work with the data without corrupting the original value.

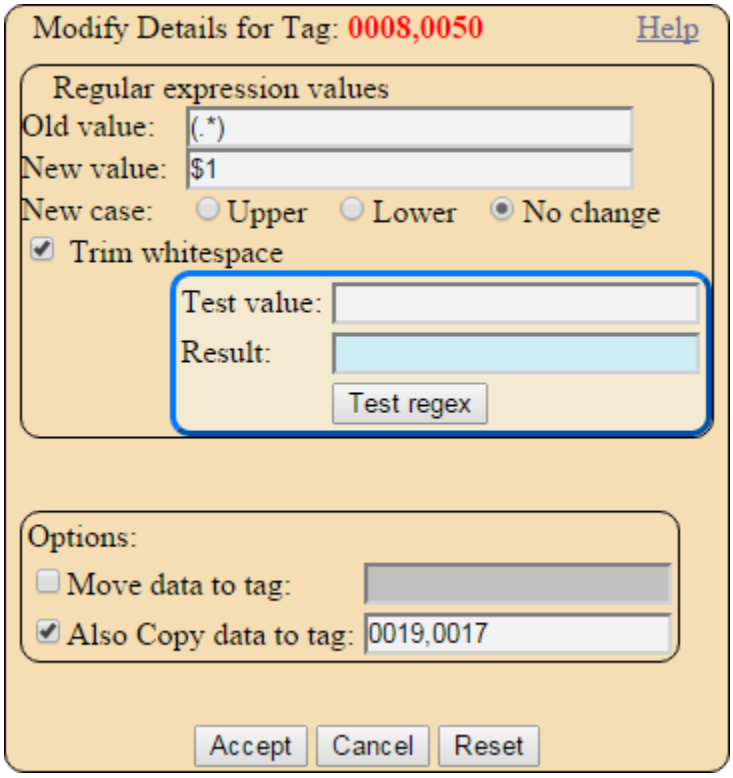


Figure 89 – Copying the data to a temporary tag

2. Create another Modify Filter. Set the Elements to Modify to be your temporary tag (0019,0017 in this example). You want to check only the first three characters of the value, so Old Value would be "`^(...).*$`". (The caret indicates that the string has to be at the start; the three dots in the parentheses indicate a group of three characters; the dot-star-dollar-sign indicates all the characters to the end of the value.) New Value would be "`$1`", indicating the first grouping of characters. Check "*Move Data To*" and enter the value of another temporary tag (say, 0019,0019).

   This extracts the desired characters from the value and puts them in another tag that can be compared.

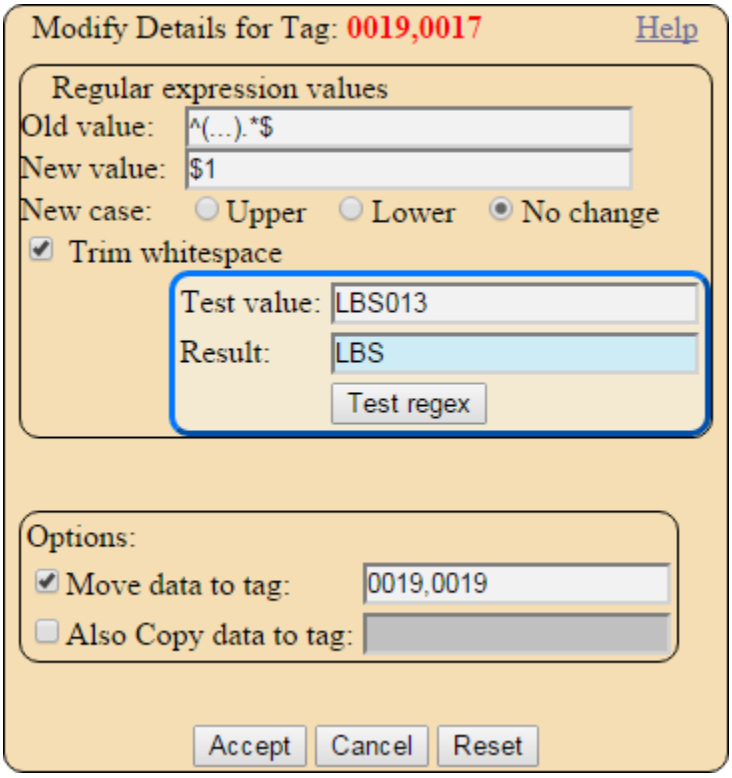


Figure 90 – Extracting the desired characters

3. Create an Add/Replace Filter. Set the Elements to Match to be our temporary tag, 0019,0019; set the value to match to be "LBS". Set the Elements to Replace as "0008,1030" for the tag and "Laurel Bridge Test Study" as the value. (See **below**.)

   This will change the Study Description value if our temporary tag has the value "LBS."

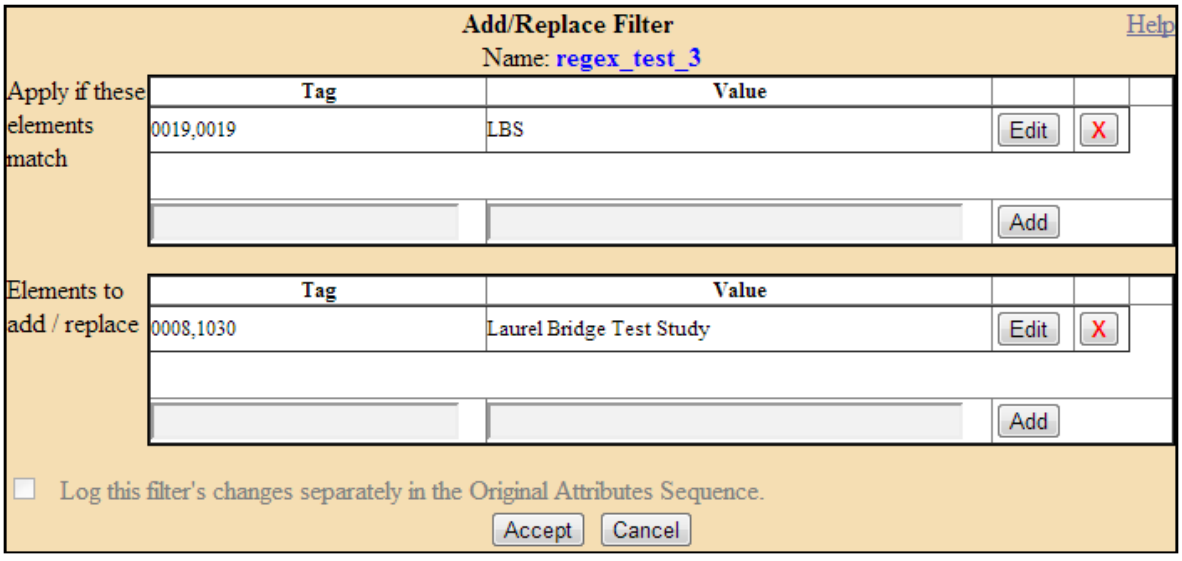


Figure 91 – Comparing the temporary tag's value

4. Create a Remove Filter. Set the Elements to Remove to be 0019,0019.

This will clean up the extra data we created with our temporary tag.

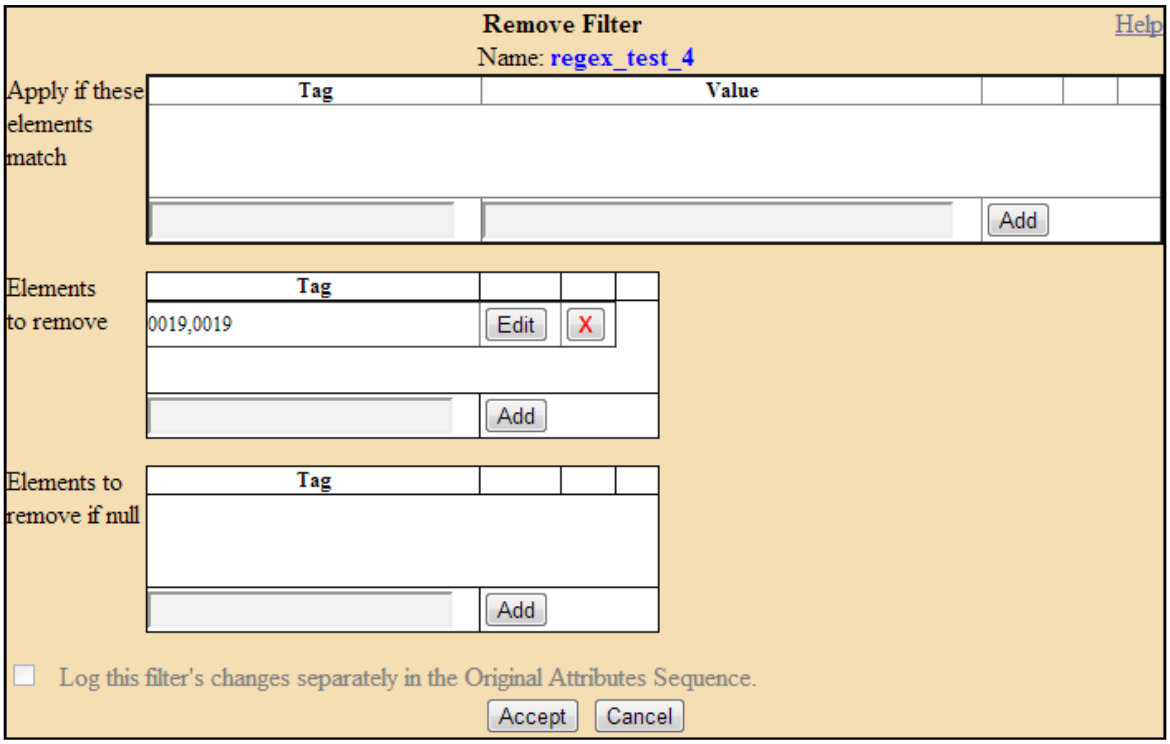


Figure 92 – Cleaning up the temporary tags

It's a little complicated at first glance, but this just shows the power of the DICOM Switchboard and its Modify Filter and the many things you can do with it. And you can add additional complexity to make sure that the filters are doing precisely what you want – for example, you could specify in each of these filters that it should be applied only if the modality was a certain type, or other such conditions.

You would use similar filters if you wanted to check that a tag exists but has no value: copy the data to a temporary tag; set the value of the temporary tag to be the existing value plus a string (e.g., "FLAG"); check if the value of the tag is only "FLAG" and do whatever action you need for the null-valued tag; and delete the temporary tag.

You can do something similar if you want to check that a tag does not exist: use a Composer filter to copy the entire value of a tag to a private tag; then use an Element Filter and check if "elements to match" equals "${1.1}" and create the new tag with the desired value, and then have a third filter to delete the private tag. The configuration for such a set of filters might look like this:


```
[ 1 ]
filter_type = DICOM_ELEMENT_COMPOSER_FILTER
[ 1/inputs ]
# Copy tag 0010,0021 to private tag.  This will get the value "??" if
# the tag does not exist.
[ 1/inputs/1 ]
tag = 0010,0021
regex = (.*)

[ 1/outputs ]
[ 1/outputs/1 ]
tag = 0077,0077
value = ${1.1}

# If the private tag has a value that evaluates to "??", set 0010,0021
# to have our desired value.
[ 2 ]
filter_type = DICOM_ELEMENT_FILTER

[ 2/elements_to_match ]
0077,0077 = ${1.1}
 [ 2/elements_to_replace ]
0010,0021 = Doe^John

# Delete the private tag
[ 3 ]
filter_type = DICOM_ELEMENT_FILTER
[ 3/elements_to_remove ]
tag = 0077,0077
```

Note: you must be careful in choosing the values of the temporary private tags that you use for something like this, to ensure that you will not overwrite a DICOM tag or a private tag. The DICOM standard requires that all private tags be odd numbered and specifies very specific rules about how they should be chosen. You may also need to add the temporary tags you choose to use to the Switchboard's Extended Data Dictionary so that the correct value representation (VR) is chosen by the underlying components.

Future versions of the DICOM Switchboard are planned to support other mechanisms that do not require the use of private tags, for managing this sort of activity.

Appendix B: Example: Loading a new filter set

This appendix shows how to use the Import feature to load a new filter set, one that may have been received from another source (e.g., LBS tech support) or that may have been created on another system, into the DICOM Switchboard configuration. Typically filters are created directly in the DICOM Switchboard application, but this approach allows filters to be created in one Switchboard context and then be imported into another DICOM Switchboard.

1 Importing the filter set

You will have received a file containing a new or revised filter set for use with the DICOM Switchboard software from some source.

1. You need to copy the desired filter set file onto your DICOM Switchboard system. Choose the "Import filter set" option from the Windows Start menu:

   `Start → All Programs → LAUREL BRIDGE SOFTWARE → Switchboard → Import filter set`

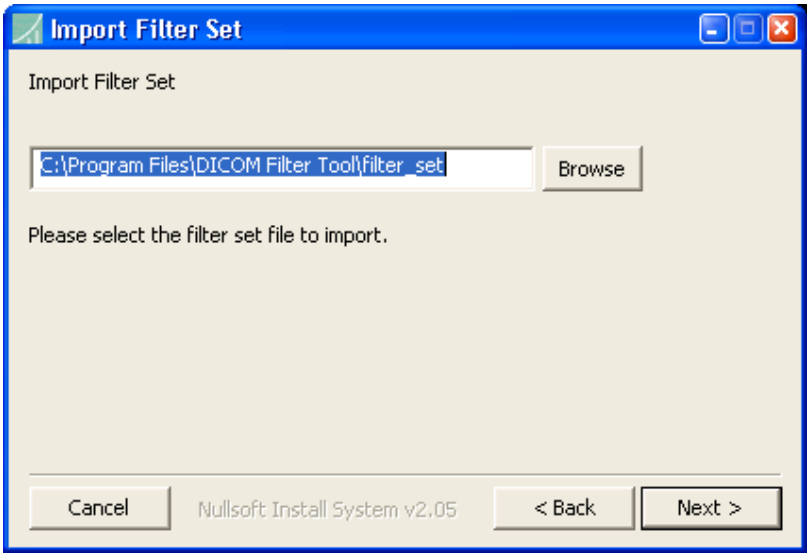


Figure 93 – Importing a Filter Set

2. Select the file for the filter set that you want to import into the DICOM Switchboard. You may use the "*Browse*" button to locate the file. (Note that the filter set to import should not have a name that is already in use in the DICOM Switchboard.)
3. Click the "*Next*" button to copy the file into the DICOM Switchboard.
4. You may need to restart the DICOM Switchboard before you are able to access the new filter set.

Running this utility copies the filter set into the appropriate directory. Typically this is:

`C:\LB Switchboard\cfg\dicom\filter_sets`

(**For Linux:** This importation is done manually by copying the file for the filter set to the `cfg/dicom/filter_sets` subdirectory under the Switchboard's installation directory. Note that you may still need to restart the DICOM Switchboard before you can use the new filter set.)

1.1 Editing the imported filter set

You may access the newly loaded filter set from the "Edit Global Filter Sets" menu on the Switchboard Operations window. Selecting the newly loaded filter set will enable the "*Edit Set*" button as shown **below**.

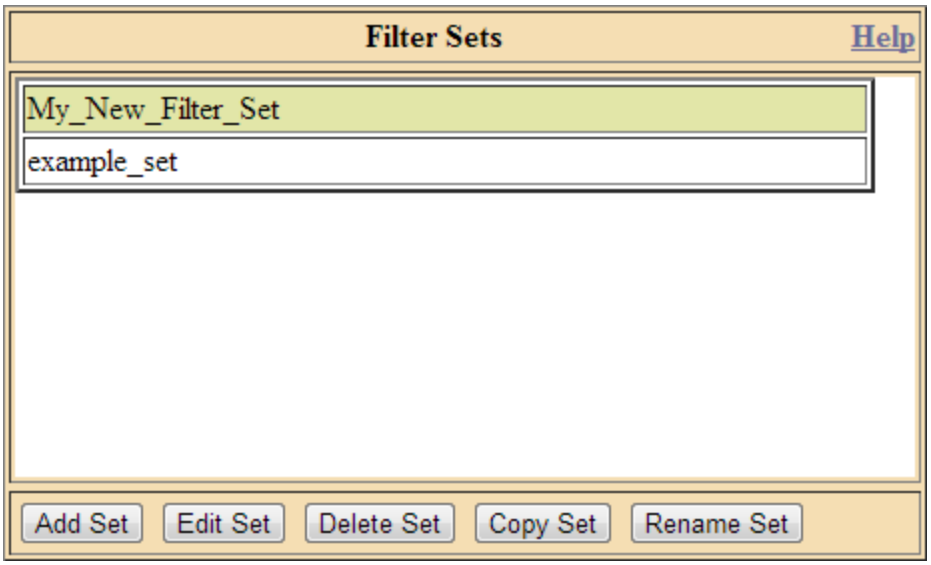


Figure 94 – Filter Set Editor Selection screenshot

1.2 Using the imported filter set in an existing Mapping Rule

Once you have the new filter imported, you have to select this filter to enable its use with your mapping rule. To do this:

1. If the DICOM Switchboard is running, you may have to Stop, then Start the Switchboard from the Switchboard Operations window to cause the new filter to become available for selection.
2. Next, from the Switchboard Operations window, select "Edit Mapping Rules".
3. Select your Rule Group and your Mapping Rule from the table at the top of the screen.
4. Click on the drop-down list next to "*Filters: To Source*" or "*Filters: To Dest*" near the bottom of the window, and select the filter set to use.

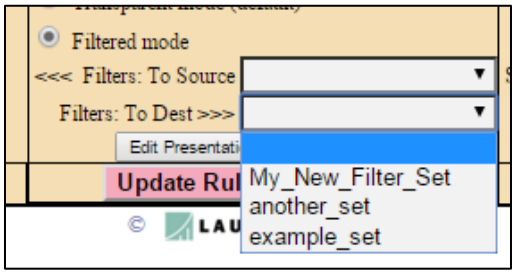


Figure 95 – Selecting an imported filter set to use

5. Accept the changes and save the modified rule.
6. Stop, then Start the DICOM Switchboard from the Switchboard Operations window to cause the new filters to be loaded.
7. Test the new filter set as required.

Appendix C: RLE Transfer Syntax Support

Is it possible for the Switchboard to support the RLE (1.2.840.10008.1.2.5) transfer syntax?

The answer is "Yes", although with some caveats. The Switchboard can support RLE but cannot create multi-frame RLE images. Since the ability to create RLE multi-frame datasets is not available, the RLE transfer syntax is not listed among the supported transfer syntaxes in the application's user interfaces.

If your application of the Switchboard product can handle the lack of ability to create multi-frame RLE datasets, then you may manually enable the RLE transfer syntax as described below.

Manually modifying the dcf_switch's configuration file to enable RLE support:

For incoming RLE, you can tell the Switchboard and its GUIs that RLE is supported by editing the `dcf_switch` configuration file. To locate this file, see:

```
C:\LB Switchboard\cfg\apps\defaults\dcf_switch
```

Edit the file using a text editor and find the section:

```
[ java_app/dcf_switch/supported_transfer_syntaxes ]
```

At the end of that section, add the UID for the RLE transfer syntax (1.2.840.10008.1.2.5), for example:

```
# RLE
transfer_syntax = 1.2.840.10008.1.2.5
```

Save the modified configuration file and then restart the Switchboard application.

After making this change the Switchboard can take RLE in, and can send RLE out provided it is not a multi-frame dataset. In addition, as long as the data is not multi-frame data, it may also be filtered by the Switchboard.

Appendix D: Renaming your computer

When you install the DICOM Switchboard, it queries the operating system to find the name of the computer it is being installed on and then configures itself to run on that box. But sometimes you may find it necessary to rename your computer *after* you have already installed the DICOM Switchboard, and then you may find that the DICOM Switchboard doesn't work at all. If you are in this situation, it is simple to reconfigure the DICOM Switchboard to use the new computer name.

1. Edit the file **dsb_vars.bat**; this will be located at the top of the directory where you installed the DICOM Switchboard, e.g., `C:\LB Switchboard`. In the file, change the HOSTNAME value from the old name to the new name, and then save the changes.
2. Under the Switchboard's installation directory, go to the subdirectory `httpd/html`. Delete the files `Auto_redirect.html` and `dcf_redirect.html`.
3. Go the `httpd/conf` subdirectory and delete the file `httpd.conf`.
4. Open a Switchboard Command Prompt, and verify that the HOSTNAME environment value has changed to the new value. You can do this by running the "`set hostname`" command and observing that the correct value is returned.
5. From the Switchboard Command Prompt, run the command "`perl –S config_apache.pl`".

The files that you deleted will be recreated with the correct hostname in them, and your DICOM Switchboard is ready to go again!

For Linux: Do *not* delete the file `<instdir>/httpd/conf/httpd.conf`; instead, edit it and change the host name as necessary.

1 Changing the Web Port

If you wish to change the port that the Apache server uses, do the same steps as above, except that you will replace the value for DCF_APACHE_PORT instead of HOSTNAME. And in Step 4 above, run the command "`set DCF_APACHE_PORT`" instead to check that the new value is correct.

Appendix E: Editing the Extended Data Dictionary

The DICOM Switchboard allows you to add tags to the DICOM data dictionary that it uses – this allows you to specify the VR (Value Representation) and VM (Value Multiplicity) for private tags or for new tags that older versions of the Switchboard may not have included in the built-in dictionary.

If the Switchboard does not know the VR of a tag, such as in the case of a private element in a dataset in implicit format, then that element is treated as if it were VR: UN; this causes the value found to be interpreted as a binary value instead of as a string – for example, the string "Bob", when interpreted as binary data, is stringified to the hexadecimal bytes: "0x420x6F0x620x20". To prevent this treatment of the data, you must add any private tags you plan to process (filter, copy, add, replace, etc.) to the Extended Data Dictionary – doing this will allow the Switchboard to lookup the VR of the tags to know how to correctly interpret and process the element contents.

To edit the Extended Data Dictionary, from the Switchboard's home page, select "DCF: Advanced Options". In the lower left corner of the DCF page you'll find "Edit Extended Data Dictionary". Clicking that option will bring you to the web pages for editing the Extended Data Dictionary (note that only the default dictionary – `$DCF_ROOT\cfg\dicom\ext_data_dictionary` – can be edited via this page).

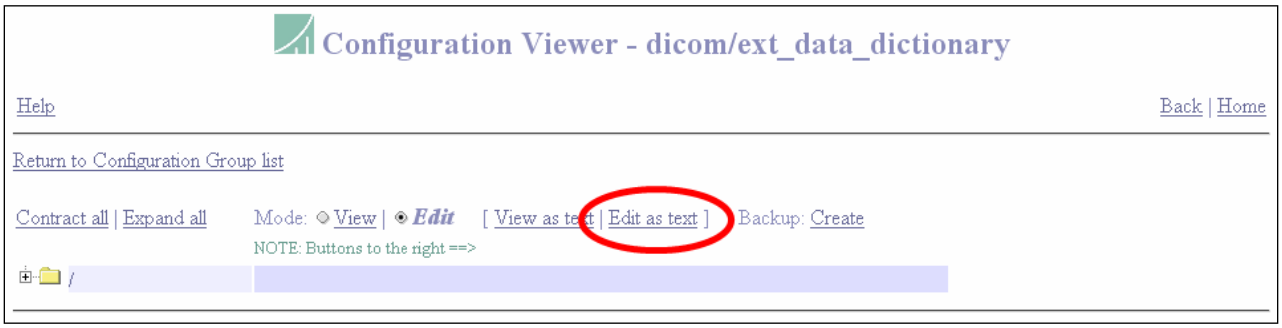


Figure 96 – Editing the Extended Data Dictionary

Two options are available for editing – text editing or editing via a GUI.

A user can edit it as text by clicking the "Edit as text" link near the top of the page.

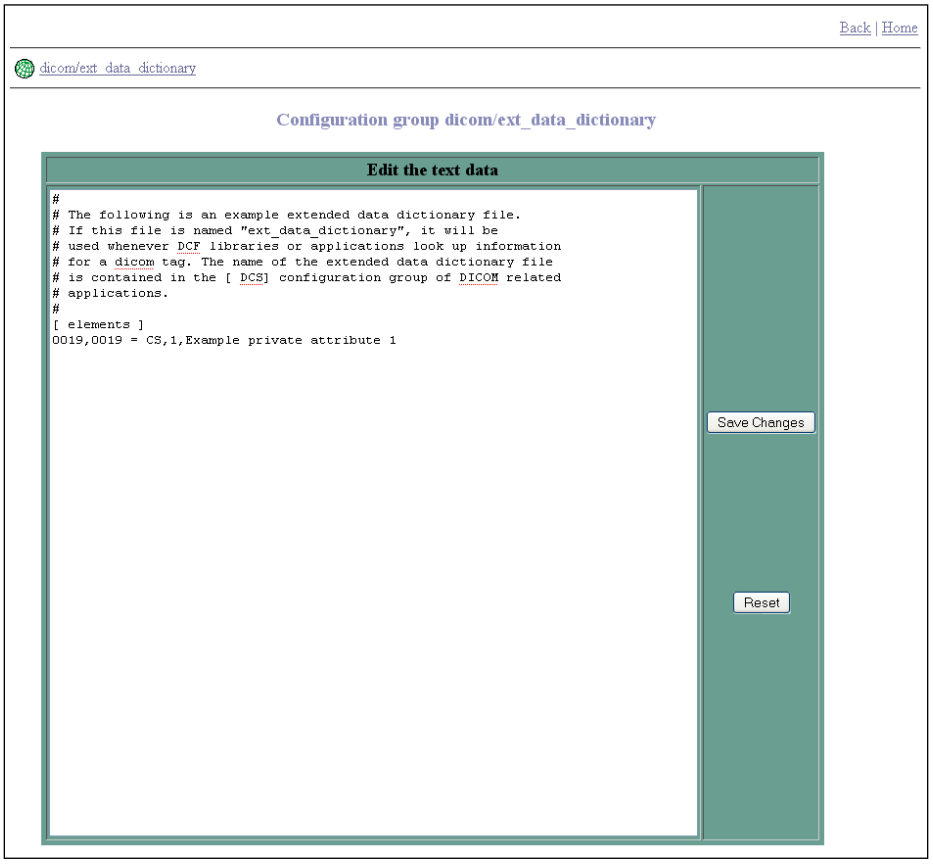


Figure 97 – Editing the Extended Data Dictionary as text

Or a user can add attributes to the dictionary via the GUI – to use this approach, expand the top group by clicking on the slash, then choose the "elements" group by clicking its name.

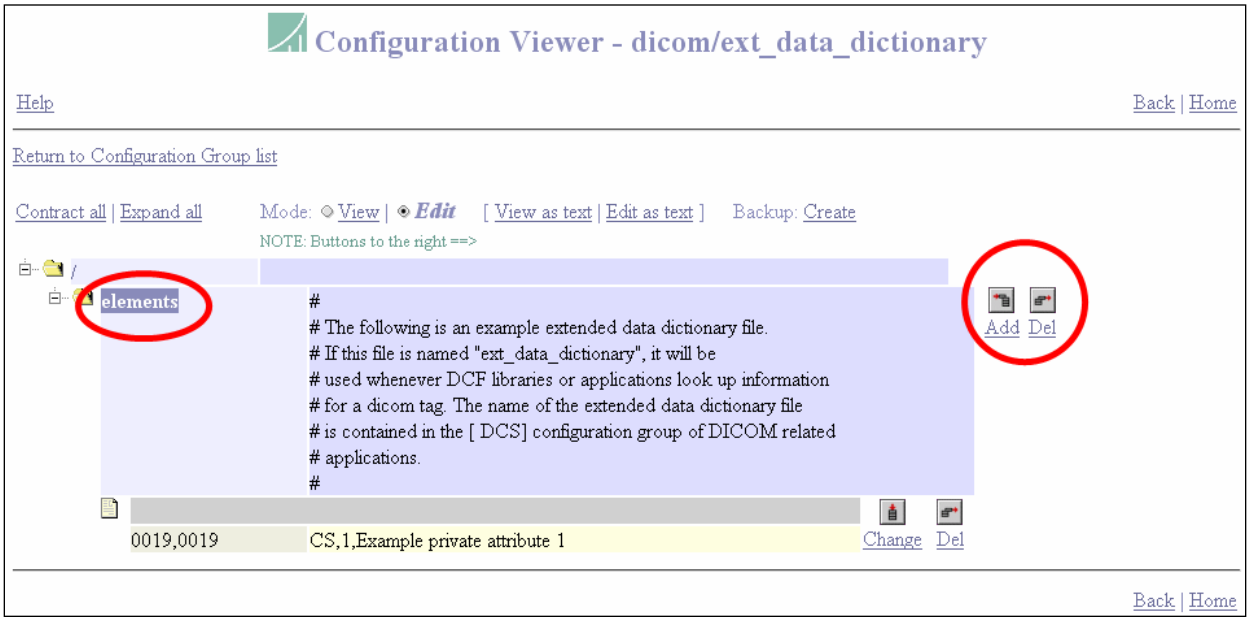


Figure 98 – Adding to the Extended Data Dictionary

To add elements, click the "Add" button (at the far right) that is at the same level as the Elements group. This will open a new page where you can add new groups or attributes. For the Extended Data Dictionary, you want to enter values in the "attribute" boxes – these are highlighted in **orange** in the illustration **below**. Enter the appropriate values in the boxes, and click "Submit".

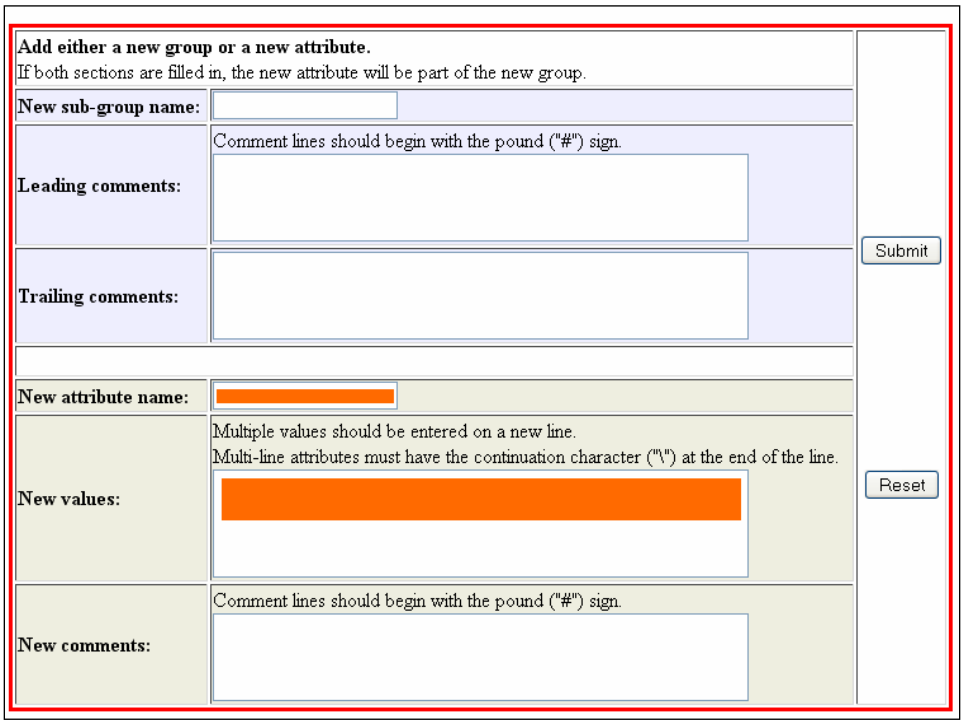


Figure 99 – Enter the new values for the data dictionary

The values you need to enter are as follows: the tags are entered as new attribute names, and then the values for the attributes are the VR, VM, and a description, separated by commas.

For example: To add tag 0029,1020 to the data dictionary.

In the **new attribute name** field
enter "0029,1020".
In the **new values** field,
enter "CS,1,Example private attribute 1".

This **new values** field text means
"the VR of the tag is CS, the VM is 1, and the description is "Example private attribute 1". This means that tag 0029,1020 will be interpreted as a Code String, and there can be only 1 value. (Consult part 5 of the DICOM Standard for details on any limits of each VR.)

Repeat this process as necessary for additional tags.

When you are done adding tags to the Extended Data Dictionary, click the "Home" link at the top right corner to return to the Switchboard's Home page. You will have to restart the Switchboard before you can use the new values in the Extended Data Dictionary.

Appendix F: Additional Information on Apache

The DICOM Switchboard has been designed to use Apache 2.2.16. This web server allows the Switchboard to be configured easily via web pages and Java applets. Once the configuration is done, you can leave the Switchboard running but are not required to have the web server running as well – this ensures that the configuration cannot be changed by a user connecting to the Switchboard's web server either accidentally or maliciously – this makes the Switchboard's configuration more secure.

Previous versions of the Switchboard used Apache 1.3.33, which allowed the web server for the DICOM Switchboard to be controlled from a Windows command prompt, stopping and starting it via the Perl scripts included with the Switchboard. Later versions of Apache (2+) are designed to be run as a service by default and so do not provide that functionality – you cannot easily stop Apache2 running in a console, for example. This made it much more complicated for you to run the web server only when you want it to be run, instead of letting it run all the time – for example, the `kill_apache.pl` script does not work on Windows with Apache 2. But Apache 2 provides many security enhancements and additional features, so the Switchboard has been modified to use Apache 2 by default. You can still use Apache 1.3.33 if you so desire.

1 Configuring Apache to run as a Service

If you want to have the web server running all the time, you can select that option when you install the DICOM Switchboard. You can also modify the existing `httpd.conf` file – located by default in the directory `C:\LB Switchboard\httpd\conf` – and configure Apache to run as an auto-starting service.

1. To configure Apache as a service, run this command:

   ```
   httpd.exe –k install –n "DSB Apache"
         –f "<Switchboard install dir>/httpd/conf/httpd.conf"
   ```

   This will install Apache as a service with the name "DSB Apache" and tell it to use the specified configuration file.

2. Edit the httpd.conf file to set the DCF environment variables. Change the "PassEnv" commands at the end of the file to be "SetEnv" and to have the corresponding values from a Switchboard Command Prompt. Do this for all of the DCF environment variables, the Java Classpath, and the Path environment variable, too.

   Example: Change '`PassEnv DCF_LIB`' to

   '`SetEnv DCF_LIB "C:/LB Switchboard/lib"`'

   Note that you may need to enclose the paths in quotation marks if there are any spaces in the path.

3. You may also need to set the DCF_CFG environment variable in the System environment so that it is accessible to the web server when it starts. This can be done by right-clicking on "My Computer," selecting the "Advanced" tab, clicking on the "Environment Variables" button, and adding or editing DCF_CFG in the "System variables" list.

Running Apache as a Service on Linux

On Linux, to run Apache as a service, you need to create a startup script (usually in `/etc/rc.d` or a similar directory). The script should set the necessary DCF environment variables (these can be found in the `dcf.env` file in the Switchboard's installation directory) and export them. To run Apache, use the command "`run_apache.pl`". To stop it, use "`kill_apache.pl`".

Note that SUSE 13.2 uses different commands, detailed in Section **3.5.1 Configuring the Switchboard service on Linux** above.

2 Using an Alternate Web Server

So, you have decided to use a web server other than Apache 2.2.16 (the version included with the DICOM Switchboard on Windows). This is done by installing the alternate web server and then modifying its configuration file to match the configuration values in the Switchboard's default `httpd.conf` file – the setting and passing of environment variables, the document root, the use of Perl scripts for CGIs, etc. Then the web server must be started and configured to use the updated configuration file. If you want to use the existing `run_apache.pl` and `kill_apache.pl` Perl scripts, these should be modified to refer to the executable for your new web server. (Note that the `kill_apache.pl` script does not work with Apache 2.*; instead, you will have to use CTRL-C to kill Apache in the command window running it.)

If you wish to run the alternate web server as a service for the DICOM Switchboard, you will have to do the configuration steps described in the previous paragraph. You will also need to set the DCF environment variables that are required by the Switchboard, in a step similar to that shown in Step #2 above. You may also be required to install the new web server as a service.

As there are many possible web servers that can be used, it is beyond the scope of this manual to provide more detailed instructions for these steps. A good place to start is simply to compare the existing httpd.conf file with your server's configuration file and consult its documentation for more information.

The basic steps for configuring your web server are summarized below:

- Make sure you have Perl installed, including the CGI modules such as CGI::Util, CGI::Cookie, etc.
- Modify your configuration files to serve up the Switchboard's Perl scripts and files
  - Set the DocumentRoot to be `<INSTDIR/httpd/html>`
  - Set ScriptAlias to refer to `<INSTDIR/httpd/cgi-bin>`

    You may need to configure it to recognize the scripts as CGI; in Apache this is done as:

    "`AddHandler cgi-script .exe .cgi .dcfpl`"
  - Set the user/group that the web server should run as
  - Set or Pass the Switchboard's environment variables and their values; these include `LD_LIBRARY_PATH`, `CLASSPATH`, `PATH`, `DCF_TMP`, `DCF_LOG`, `DCF_CFG`, `DCF_LIB`, `DCF_ROOT`, `DCF_USER_ROOT`, `DCF_HTTPD_ROOT`, `DCF_APACHE_PORT`, `DCF_PLATFORM`, `HOST`, `USER`, and `PERL5LIB`; you may find the values to use for many of these in the `dcf.env` file in the `<INSTDIR>` directory.

3 Using Apache 1.3 on Linux

The DICOM Switchboard is designed to use Apache 2.2.16 to serve up its web pages, but previous versions of the Switchboard used Apache 1.3. Most recent versions of Linux ship with Apache 2.x, but it is still possible to use the older Apache if you so desire.

If you want to build Apache 1.3 to use the existing httpd configuration file without any special changes, you can download the latest sources from http://httpd.apache.org/. Extract the sources, change into the source directory, and build Apache using these (or similar) commands:

```
./configure --prefix=<apache install dir> --enable-module=most
     --enable-shared=max

make

make install
```

(Note that you may need to be root or a different user with special permissions if you want Apache to be installed in protected directories such as `/opt`.)

You will need to edit the Switchboard's `app_platforms.cfg` file and set the values appropriately before running the script `dcfsetup.pl` to set up the configuration files. Set the APACHE_ROOT value to be where you installed Apache. Then continue with the setup procedures as described in Chapter **3.4 Installing the Dicom Switchboard on** Linux.

You may need to update the directory permissions for the Switchboard if you installed it as root, since Apache usually runs as "nobody", while the Switchboard runs as whatever user you installed it as.

4 Authenticating Access to Switchboard Web Pages

As you use the DICOM Switchboard and its web pages, you will realize how powerful the web interface is and how easy it could be for an untrained user to change its settings and disrupt the operation of the Switchboard, yet you still want to have the web interface available to you for configuring and monitoring the Switchboard. In such situations, you might want to configure the Switchboard's web server to require users to "log in" before being able to access the web pages. This would restrict access to only those users that have the appropriate permissions; it is also possible to configure the access restrictions so that some users could access certain sections but not other sections.

The steps described below will allow you to require authentication to access the Switchboard's web pages.

(Please note that the steps described have been tested for Apache **1.3.33 and 2.2.16**; if you are using a different web server, the steps will be similar but not identical.)

Steps to require authentication to access Switchboard web pages:

1. Create the password file.
   This is done via the **htpasswd** utility, provided by Apache when you installed it.

   To see the usage, at a command prompt type:

   ```
   htpasswd
   ```


You must create the password file to use if it does not already exist. From **outside** the *httpd* docs tree (for example, at DCF_ROOT), run

```
htpasswd -c %DCF_ROOT%\dcf_passwords <username>
```

This will create the password file "**dcf_passwords**". (The file should be created outside the web server root so that it is not accidentally served up by the web server.)

You will be prompted for a password for the user, and required to type it twice. Once the password file exists, you should omit the "-c" flag. (The "-c" flag is only for creating the file the first time; if you use it and the file exists, then you will create a new file, overwriting the existing one, and you will lose any user info in the original file.)

For subsequent users that need to be added or to change the passwords of existing users, run the command

```
htpasswd %DCF_ROOT%\dcf_passwords <username>
```

(Note that the "-c" option is not used.)

2. Configure Apache to use the file for authenticating users.
   Create an "**.htaccess**" file in the directory you wish to protect. This will primarily be the `%DCF_ROOT%/httpd/cgi-bin` directory and the `%DCF_ROOT%/httpd/html` directory.

   The contents of the file would look something like this:

```
AuthType Basic
AuthName "Access restricted to authorized DICOM Switchboard users"
AuthUserFile "C:/LB Switchboard/dcf_passwords"
Require valid-user
```

   This specifies that authentication is required and that the dcf_passwords file should be checked for the authorized usernames and passwords. Note that you can change the **AuthName** value to whatever text you want to be displayed in the password entry dialog box that will pop up when a user is required to log in. Also, please note that you may need to put the path to the dcf_passwords file in quotes if there are spaces in the path.

3. Modify the httpd.conf file (in `%DCF_ROOT%/httpd/conf`) to allow the .htaccess file to override some options.
   You should do this for each "<Directory>" section in the httpd.conf file that has a directory you wish to protect. To restrict access to the CGI scripts, find the "<Directory>" section for the **cgi-bin** directory (`%DCF_ROOT%/httpd/cgi-bin`) and change **AllowOverride** from **None** to **AuthConfig**. You may need to do the same thing for the `%DCF_ROOT%/httpd/html` directory.

4. Stop, then restart the Apache server.

   Stop Apache by pressing CTRL-C in the command prompt that is running Apache, or use the Windows Task Manager to stop the httpd.exe. (**Linux**: Use the `kill_apache.pl` script.)
   This will stop the Switchboard's Apache web server. Restart the server by running the command

```
perl run_apache.pl
```


For more information about authenticating users and additional measures you can use, see

http://httpd.apache.org/docs/2.2/howto/auth.html

For example, this page can show you how to configure the Apache web server so that it can be accessed only from your internal network.

Appendix G: Customizing the Switchboard

1 Customization

The Switchboard has been designed to provide ease of use and readability, and it is designed to run by itself as a web-accessible application. There may be times when you wish to customize it in some way or integrate it with your own web utilities in some way, for these cases you should request the Tech Note: Switchboard Customization & Integration from Laurel Bridge Software.

2 Disclaimer

If you elect to modify the distributed Switchboard application in any of these ways, then you assume the responsibility for the modified software and any changes that you implement. Laurel Bridge Software may be able to provide assistance with such integration efforts, but this assistance will be outside of standard application support and will be at extra cost.

Appendix H: Using & Creating Many Mapping Rules & Filters

There are some situations where you may want to use the DICOM Switchboard to handle many, many associations at one time and you will need hundreds – possibly even thousands – of mapping rules and filters. Unfortunately, the DICOM Switchboard does not come with a method for easily generating these thousands of rules – there are just too many options! But while there is no simple user interface for creating these rules and filters, it is a process that can be done manually without much difficulty.

1. Open the Mapping Rule Editor GUI. Delete all of the existing mapping rules. Create a new mapping rule group and a single mapping rule in it; this includes creating any filter sets that made be needed (see the note on this below). This will give you a clean slate to help you understand how the mapping rules are formatted.
2. From the Switchboard's Utilities menu (on the Windows Start menu), select the utility to "Import and Export Mapping Rules". Select the option to export the mapping rules and specify a name for the output file. (Section **12.9 Import and Export Mapping Rules** has more information on the Import/Export utility.)
3. Examine the output file – this is a plain text file that can be edited in vi or Notepad – so that you understand what fields go where and what they are named. Create a script to create all the mapping rules that you need – make sure that each rule has a unique name, and save the results in a single text file.
4. Use the "Import / Export" utility to import the new file of mapping rules into the Switchboard.
5. You will have to restart the Switchboard before you can use the new mapping rules. You may also have to use the Mapping Rule Editor to activate your new mapping rule set.

For the filters, you will probably want the same filters to be used by all the mapping rules. The easiest way to do this is to have the filters that are used stored in a shared filter set (this is a single file). In this case, you would want your mapping rules (from the steps above) to specify the filter set use. So, all of the mapping rules would share one filter set.

You would want to do a similar process to create multiple filters.

1. Use the Filter Set Editor GUI to create your filter.
2. Examine the filter set file to understand the fields and what goes in each one. The filter sets are saved in the files in `$INSTALL_DIR\cfg\dicom\filter_sets`. These files are plain text files that can be edited with Notepad or vi.
3. Create a script to make all the different filters you need and save the output in a single text file (this file will contain all of the filters).
4. Put the new filter set file in the directory `$INSTALL_DIR\cfg\dicom\filter_sets` and restart the Switchboard, or you can use the "Import Filter Set" option from the Utilities menu (this utility doesn't require that the Switchboard be restarted).

(See **Appendix B: Section 1 Importing the filter set** for more information on the Import Filter Set Utility.)

One additional note on the filters: In our experiences with clients using the Switchboard, they have very rarely needed to use thousands of individual filters when changing a value. Instead, they have been able to use the Mapping List Filter to do the substitutions that they wanted. But this can depend on what you want the filters to do – if you are substituting value A for value B, a Mapping List Filter can do this. On the other hand, if you need to parse the value and rearrange the value with regular expressions, a Mapping List Filter can **not** do this – this would be done best by a single Modify Filter or by the Composer Filter.

Appendix I: Conditional Filters

Conditional Filters provide a more advanced way to filter DICOM data – you set a number of *conditions* and then choose the *actions* to be performed if those conditions are met. Switchboard does not (yet) provide a UI for creating and editing Conditional Filters, but you can create them yourself and incorporate them via a standard text editor such as Notepad or GVIM. The basic conditions and actions are described below; if you need additional information, contact Laurel Bridge Software for assistance (support@laurelbridge.com).

1 Conditions

- com.lbs.DCS.DicomTagCondition

A Condition is defined as the DICOM tag whose value is to be checked, an operator, and the value that the tag should match. The operators supported are:

- EQUALS
- NOT_EQUALS
- CONTAINS
- NOT_CONTAINS
- STARTS_WITH
- ENDS_WITH
- REGEX_MATCH
- REGEX_MULTI_MATCH
- EMPTY
- NOT_EMPTY
- LESS_THAN
- GREATER_THAN
- LESS_THAN_OR_EQUAL_TO
- GREATER_THAN_OR_EQUAL_TO
- EXISTS
- NOT_EXISTS

Note a VALUE is required for all operators except the following:

- EMPTY
- NOT_EMPTY
- EXISTS
- NOT_EXISTS

Multiple Conditions can be used by grouping them together and specifying if all conditions must apply (AND) or if any condition must apply (OR). See the example below to better understand how to define a Conditional Filter.

2 Actions

Actions have different attributes that define what DICOM tags are operated on and how they are used. Below is a list of the supported Actions and their attributes.

- **com.lbs.DCS.CopyTagAction** – Copies the value in one tag to another tag, overwriting the destination if it exists.
  - copy_from_tag = XXXX,XXXX
  - copy_to_tag = XXXX,XXXX
- **com.lbs.DCS.ChangeCaseAction** – Changes the case of the specified tag to uppercase or lowercase. Ignores non-string type tags.
  - tag = XXXX,XXXX
  - case = [UPPERCASE|LOWERCASE]

- **com.lbs.DCS.InsertOverwriteTagAction** – Inserts a new DICOM tag with the supplied value.
    - tag = XXXX,XXXX
    - value = XXXXXXX
- **com.lbs.DCS.ModifyTagAction** – Modifies the tag specified using the given regular expression and substitution string.
    - tag = XXXX,XXXX
    - pattern = X
    - substitution = X
- **com.lbs.DCS.MoveTagAction** – Action that moves the value in one tag to another tag, overwriting the destination if it exists.
    - move_from_tag = XXXX,XXXX
    - move_to_tag = XXXX,XXXX
- **com.lbs.DCS.NewUidAction** – Action that inserts a new DICOM UID at the specified tag location overwriting existing values if they exist.
    - tag = XXXX,XXXX
- **com.lbs.DCS.PadTagAction** – Pads the tag specified with the given character up to the desired length, padding from the left or the right.
    - tag = XXXX,XXXX
    - pad_character = X
    - desired_length = X
    - pad_direction = [LEFT|RIGHT]
- **com.lbs.DCS.RemoveTagAction** – Action which removes the specified tag. Can optionally only remove if the tag is empty.
    - tag = XXXX,XXXX
    - only_if_empty = [TRUE|FALSE]
- **com.lbs.DCS.TrimTagAction** – Trims the tag value specified from the front, end, or both.
    - tag = XXXX,XXXX
    - location = [LEFT|RIGHT|BOTH]
- **com.lbs.DCS.FindReplaceTagAction** – Performs a Find/Replace operation on the original tag value optionally using Regular Expressions.
    - tag = XXXX,XXXX
    - find = X
    - replace = X
    - regex = [TRUE|FALSE]
    - ignore_case = [TRUE|FALSE]
- **com.lbs.DCS.PrependAppendTagAction** – Prepends or appends the tag specified with the given string.
    - tag = XXXX,XXXX
    - value = X
    - location = [LEFT|RIGHT]
- **com.lbs.DCS.MappingListAction** – Applies a Mapping List transformation to the input dataset. See Section **8.4.1 Mapping List Filter** for details.
    - match_tag = XXXX,XXXX
    - match_tag = YYYY,YYYY
        - 1 or more of these are required
    - replace_tag = ZZZZ,ZZZZ
    - replace_tag = VVVV,VVVV

        - 1 or more of these are required
    - mapping_cfg_name = <filename>
        - This is the complete path to a file defining the mappings to search.
    - no_match_option = [0|1|2]
        - 0 = reject the data set by aborting the association; the data set is not forwarded to the destination
        - 1 = log a warning message and forward the filtered data set to the destination
        - 2 = ignore the error and forward the filtered data set to the destination
    - mapping_cfg_format = CSV
    - mapping_cfg_delimiter_char = ","
        - This can be any character, but it must be enclosed in quotes; typical characters are commas, tabs, and semi-colons.

3 Example Conditional Filter

```
[ sample_filter ]
filter_type = DICOM_CONDITIONAL_FILTER
create_original_attribute_seq = false
save_pixels_in_oas = FALSE

[ sample_filter/conditions ]
mode = AND
[ sample_filter/conditions/0 ]
class_name = com.lbs.DCS.DicomTagCondition
[ sample_filter/conditions/0/data ]
tag = 0008,1120.0008,1155
operator = EXISTS
value =
[ sample_filter/conditions/1 ]
class_name = com.lbs.DCS. DicomTagCondition
[ sample_filter/conditions/1/data ]
tag = 0008,1120.0008,1155
operator = REGEX_MATCH
value = ^1\.2\.840\.10008\.3\.1\.2\.3\.[\s]*$

[ sample_filter/actions ]
[ sample_filter/actions/0 ]
class_name = com.lbs.DCS.InsertOverwriteTagAction
[ sample_filter/actions/0/data ]
tag = 0991,ae04
value = 1.2.840.10008.3.1.2.3.
[ sample_filter/actions/1 ]
class_name = com.lbs.DCS.RemoveTagAction
[ sample_filter/actions/1/data ]
tag = 0008,1120.0008,1155
value =
```

This example shows how to define multiple conditions and multiple actions. The top-level group – **shown in red** – specifies that this is a Conditional Filter and how the Original Attributes Sequence (OAS) should be handled. The **conditions** – **shown in blue** – specify two conditions, both of which must be true ("`mode = AND`") in order for the actions to be performed. The conditions specify that a tag in a sequence must exist and that it must match the given regular expression. The two **actions** – **shown in green** – specify that tag 0991,ae04 be given a certain value and that tag 0008,1155 in sequence 0008,1120 be removed. Note that the configuration for each action specifies the full class name of the implementation of the action before the data for the action is defined.

Appendix J: Specialized Filtering

Switchboard can be configured to do highly-specialized filtering operations. One example is that you can tell Switchboard not to forward certain requests or even to drop certain responses.

- Drop Message mode – To drop a message, your filter should change the message's command code to the value 0x0. Switchboard will then interpret the data and decide to drop the message entirely.
- Loopback Echo mode – For example, in this mode, a message received from an SCU will be modified from a C-Store-RQ to a C-Store-RSP and sent back to the SCU, but nothing will be sent to the SCP. To do this, your filter should change the DIMSE message's Command Field from RQ to RSP – this is done by OR'ing the original value with 0x8000; Switchboard will then interpret the data and send the response back to the original sender, with nothing being sent to the original receiver. (Consult the DICOM Standard for the appropriate values to use.)

These kinds of filters can be used, for example, if you want your SCU to be able to send all types of data ***to*** Switchboard but want only certain types actually to go ***through*** Switchboard to the destination. For example, you may want only MG studies to be sent through but find it too difficult to configure your SCU to send only MGs – create a filter for Loopback Echo Mode with the appropriate conditions, and any non-MG studies will not go through Switchboard, but the proper RSP response messages will be sent back to the SCU.

Or if you are doing C-Finds but want only studies related to certain hospitals to be reported back to the Query SCU. You could configure a filter for Drop Message mode, with matching conditions such that it is applied only if the Institution Name does not match a desired value – any C-Find-RSPs that do not match will be dropped, and the Query SCU will never know that not all results were returned to it.

Note that these filters must be carefully configured as to when they are applied – you may want to drop certain results, but you still want the "operation completed" message to be returned.

Any kind of filter can be used for these operations – you may set up an Element Filter (see Chapter **8 Filters and Filter Sets** above) through the GUI, or you may create **Conditional Filters** manually.

If you need help with these modes, contact Laurel Bridge Software for assistance (support@laurelbridge.com).

Appendix K: Session Settings

```
#
# Example of complete DicomSessionSettings override configuration file:
# Any subset of these attributes can be present in a configuration file and will be
# used as overrides to the default settings.
# The default values for DicomSessionSettings for the Switchboard application
# can be found in <Switchboard_Install_Dir>/cfg/apps/defaults/dcf_switch
# in the group labeled: [ java_lib/DCS/default_session_cfg ].
#
debug_flags = 0
max_read_pdu_size = 32768
max_write_pdu_size = 32768
decode_un_seqs_in_ile = YES
ignore_max_length_negotiation = NO
favor_proposed_syntax_order = YES
pdu_read_timeout_seconds = -1
pdu_write_timeout_seconds = -1
send_dimse_timeout_seconds = -1
receive_dimse_timeout_seconds = -1
stream_mode_buffer_size = 16384
pdu_write_delay_seconds = 0
pdu_read_delay_seconds = 0
association_idle_timeout_seconds = -1
associate_response_timeout_seconds = 30
release_response_timeout_seconds = 5
poll_frequency_per_second = 10
pre_association_script =
post_association_script =
disable_multi_pdv_pdus = YES
input_filter_cfg_name =
output_filter_cfg_name =
remove_incoming_role_selection_items = NO
remove_outgoing_role_selection_items = NO
# if set, private tags in a dataset with VR UN that are in the data dictionary (via the extended
# data dictionary) with a different VR will be fixed to have the VR from the dictionary.
# if not set, a warning will be logged if the VRs are not the same
fix_private_tags_with_vr_UN = NO
# if set, standard tags in a dataset with VR UN that are in the data dictionary
# (via the default data dictionary) with a different VR will be fixed to have
# the VR from the dictionary.  If not set, a warning will be logged if the VRs are not the same
fix_standard_tags_with_vr_UN = NO
# if set, explicit transfer syntax codecs will back up and attempt to decode
# the element header using implicit little endian when the encoded VR is not
# one of the standard 27.  This may result in being able to read an otherwise
# broken dataset.  A warning is printed when this occurs.
fix_implicit_in_explicit = NO
scu_socket_receive_buffer_size = 0
scu_socket_send_buffer_size = 0
```



```
# size above which OB, OW, OF, UN data will be streamed to/from
# temp files rather than fully buffered.  If large_element_file_mode
# is true, the OB, OW, OF and UN non pixel data from file based
# streams will be read from the original dataset on disk when needed.
# The value 0 disables special handling for large elements.
large_element_threshold = 10000000
# Top-level pixel data is always compressed according to the output transfer syntax.
# Icon image sequences may be compressed in the output transfer syntax or uncompressed.
# The following modes apply to the Icon Image Sequence (0088,0200) pixel data elements:
# 0 = never compress
# 1 = compress if enable_compression_pass_through_mode is true and output
#       transfer syntax is same (pass through)
# 2 = compress if pass through or if output transfer syntax is compressed lossless
# 3 = always pass through or compress to output transfer syntax
icon_image_sq_compression_mode = 0
# The following modes apply to sequences besides the Icon Image Sequence that
# contain pixel data elements:
# 0 = never compress
# 1 = compress if enable_compression_pass_through_mode is true and
#      output transfer syntax is same (pass through)
# 2 = compress if pass through or if output transfer syntax is compressed lossless
# 3 = always pass through or compress to output transfer syntax
other_image_sq_compression_mode = 0
enable_compression_pass_through_mode = YES
enable_streaming_mode = YES

[ input_filters ]

[ output_filters ]

[ supported_transfer_syntaxes ]
# explicit-little-endian
transfer_syntax = 1.2.840.10008.1.2.1
# implicit-little-endian
transfer_syntax = 1.2.840.10008.1.2
# explicit-big-endian
transfer_syntax = 1.2.840.10008.1.2.2
# JPEG lossless prediction selector = 1
transfer_syntax = 1.2.840.10008.1.2.4.70
# JPEG2000 lossless
transfer_syntax = 1.2.840.10008.1.2.4.90
# JPEG2000 lossy
transfer_syntax = 1.2.840.10008.1.2.4.91
# JPEG lossless
transfer_syntax = 1.2.840.10008.1.2.4.57
# JPEG lossy 8 bit
transfer_syntax = 1.2.840.10008.1.2.4.50
# JPEG lossy 12 bit
transfer_syntax = 1.2.840.10008.1.2.4.51
```



```
[ DicomRLECodec ]
# if false convert RLE segments to Big Endian
# if true leave RLE segments in machine byte order
RLE_legacy_mode = false

[ DicomTSCWCodec ]
# If pin_buffers is true, an attempt will be made to
# avoid buffer copies during JNI calls to compression
# code. Some JVM's may not support pinning well, or
# may suspend GC during compression operations which
# could cause potential problems.
pin_buffers = true

[ DicomTSCWCodec/jpeg_lossy ]
# lossy compression quality : 0 to 100
compression_quality = 75
# If true, then 12 bit operations will use the 16 bit IJG library
no_12bit_lib = false
# If true, derived image elements are added to data sets
# as they are written. This includes changing Image-Type,
# and adding Source-Image-Sequence and Derivation-Code-Sequence.
# A new sop-instance-uid will be created for the output data set.
add_derived_image_fields = true
# If true, signed pixel data (pixel-representation = 1 ) will
# be allowed.
allow_signed_data = false
# For codecs that support creating multiple threads for a single
# compress or decompress operation.
max_threads = 1
# If true, the TSCWIJG codec will scan the jpeg header for the
# encoded bit depth and may override the bit depth defined by DICOM.
prescan_jpeg_header = true
# If true, the header prescan will stop once the start of frame
# tag has been processed.  If false, and df_COMPRESSION is set,
# all jpeg header items will be logged to the log stream.
stop_scanning_after_sof = true
# Sanity check the rows, columns and samples per pixel in the
# jpeg header, and throw an exception if these values are not
# consistent with the values defined by the DICOM header.
check_jpeg_dimensions = true

[ DicomTSCWCodec/jpeg_lossless ]
# If true, then 12 bit operations will use the 16 bit IJG library
no_12bit_lib = true
# Set the jpeg predictor selection value for the .57 syntax.
# If the transfer syntax is 1.2.840.10008.1.2.4.70,
# this attribute is ignored and predictor selection value
# is set to 1.
```



```
jpeg_predictor_selection_value = 6
# If true, then derived image fields are added for monochrome
# images. (Some implementations add derived fields, create
# a new sop-instance-uid, etc. even for lossless compressed
# images.)
add_derived_image_fields_for_mono = false
# If true, then derived image fields are added for color
# images. (Some implementations add derived fields, create
# a new sop-instance-uid, etc. even for lossless compressed
# images.)
add_derived_image_fields_for_color = false
# If true, signed pixel data (pixel-representation = 1 ) will
# be allowed.
allow_signed_data = false
# If true, color pixel data will be allowed. Some implementations
# don't implement lossless jpeg for color, since the RGB to YBR
# color space conversion may result in some information loss.
allow_color = true
# For codecs that support creating multiple threads for a single
# compress or decompress operation.
max_threads = 1
# If true, the TSCWIJG codec will scan the jpeg header for the
# encoded bit depth and may override the bit depth defined by DICOM.
prescan_jpeg_header = true
# If true, the header prescan will stop once the start of frame
# tag has been processed.  If false, and df_COMPRESSION is set,
# all jpeg header items will be logged to the log stream.
stop_scanning_after_sof = true
# Sanity check the rows, columns and samples per pixel in the
# jpeg header, and throw an exception if these values are not
# consistent with the values defined by the DICOM header.
check_jpeg_dimensions = true

[ DicomTSCWCodec/jpeg2000_lossy ]
# override all jasper options by using this attribute
# use back-slash to end lines for a multi-lined attribute value.
# Do not use with Aware plugin
jpeg2000_codec_options =
# Specify the compression ratio
compression_ratio = 2
# Specify the number of jpeg2000 compression levels
compression_levels = 4
# If true, derived image elements are added to data sets
# as they are written. This includes changing Image-Type,
# and adding Source-Image-Sequence and Derivation-Code-Sequence.
# A new sop-instance-uid will be created for the output data set.
add_derived_image_fields = true
#
```



```
# set max tile width: 0 means full frame size
max_tile_width = 0
#
# set max tile height: 0 means full frame size
max_tile_height = 1024
# For codecs that support creating multiple threads for a single
# compress or decompress operation.
max_threads = 1

[ DicomTSCWCodec/jpeg2000_lossless ]
# override all jasper options by using this attribute
# use back-slash to end lines for a multi-lined attribute value.
# Do not use with Aware plugin
jpeg2000_codec_options =
# If true, then derived image fields are added for monochrome
# images. (Some implementations add derived fields, create
# a new sop-instance-uid, etc. even for lossless compressed
# images.)
add_derived_image_fields_for_mono = false
# If true, then derived image fields are added for color
# images. (Some implementations add derived fields, create
# a new sop-instance-uid, etc. even for lossless compressed
# images.)
add_derived_image_fields_for_color = false
#
# set max tile width: 0 means full frame size
max_tile_width = 0
#
# set max tile height: 0 means full frame size
max_tile_height = 1024
# For codecs that support creating multiple threads for a single
# compress or decompress operation.
max_threads = 1
```


Appendix L: GDPR Notes

The European Union's (EU) General Data Protection Regulation (GDPR) is a refresh of Europe's data-protection laws that harmonizes statutes across the 28 EU member states; it became effective May 25, 2018. GDPR is a law that applies to any organization doing business in the EU or with EU-based clients. It is up to the Laurel Bridge application customer to ensure that they manage the Switchboard application and the medical imaging data processed by it in a way that is conformant to their GDPR policies and practices.

The content in this appendix describes the relevant security and privacy information associated with this application. Relative to the GDPR some key points to remember are:

- The Laurel Bridge Switchboard application is installed on virtual or physical systems that are provided, configured, and controlled by the customer, therefore Laurel Bridge Software (LBS) cannot make assertions about the privacy and security of a particular installation.
- It is up to the customer to ensure that the customer's host systems on which the application components are installed have been adequately secured.
- By virtue of using this application, Laurel Bridge Software receives no private data from the customer or the customer's clients; data remains with and under the control of the customer.
- The application does not maintain a designated record set and is not a primary repository of electronic health record (EHR) or electronic medical record (EMR) data. Data processed and tracked by the application is transient and purged after a user-configurable period of time.
- Log files may possibly contain private data associated with the medical imaging data being processed. Such files should be handled in a way that is compliant with the customer's data retention and privacy policies.

Appendix M: Glossary

This Glossary contains definitions for some common DICOM related terms.

A

Abstract Syntax – A DICOM term which is identical to a DICOM SOP Class; it identifies a set of SOPs which, when taken together, represent a logical grouping. An Abstract Syntax identifies one SOP Class or Meta SOP Class.

AE Title – See **Called** and **Calling AE Title** listed below.

Application Entity (AE) – A DICOM term for defining a particular user at a particular IP address.

Association – A DICOM term for a communication context which is used by two Application Entities that communicate to one another.

Association Negotiation – The software handshaking that occurs between two DICOM Application Entities to set up an Association.

Attribute: Each DICOM information object has its own set of characteristics or attributes. Each attribute has a name and may have a value (see IOD), depending on its category.

B

Big Endian: A term for encoding data where the most-significant byte appears first and remaining bytes follow in descending order of significance; sometimes known as "Motorola" format (see Little Endian). (The term is used because of an analogy with the story Gulliver's Travels, in which Jonathan Swift imagined a never-ending fight between the kingdoms of the Big-Endians and the Little-Endians, whose only difference is in where they crack open a hard-boiled egg.)

C

Calling (Requesting) AE Title – The name used by the receiver in a DICOM Association to indicate which Application Entity it received the data from. It is the AE Title of the AE that is initiating the transfer.

Called (Receiving) AE Title – The name used by the sender in a DICOM Association to indicate which Application Entity it wants to transmit its data to. It is the AE Title of the AE that is receiving the transfer.

C-FIND: A DICOM command in which one DICOM entity asks another DICOM entity for information matching certain criteria. A C-FIND is typically issued to a source SCP asking for a list of studies with particular study dates, modality, etc.

Conformance – Conformance in the DICOM sense means to be in compliance with the parts of the DICOM Standard.

Conformance Statement – A document whose organization and content are mandated by the DICOM Standard, which allows users to communicate how they have chosen to comply with the Standard in their implementations.

C-MOVE: A DICOM command in which one DICOM entity asks another DICOM entity to move information (typically a study) to yet a third DICOM entity.

D, E, F, G, H

Data Dictionary: A registry of DICOM Data Elements which assigns a unique tag, a name, value characteristics, and semantics to each Data Element (see the DICOM Data Element Dictionary in DICOM PS 3.6-2004).

Data Element: A unit of information as defined by a single entry in the data dictionary. An encoded Information Object Definition (IOD) Attribute that is composed of, at a minimum, three fields: a Data Element Tag, a Value Length, and a Value Field. For some specific Transfer Syntaxes, a Data Element also contains a VR Field where the Value Representation of that Data Element is specified explicitly.

Data Set – Exchanged information consisting of a structured set of Attribute values directly or indirectly related to Information Objects. The value of each Attribute in a Data Set is expressed as a Data Element.

DICOM – Digital Imaging and Communications in Medicine. A network protocol for communication between medical devices. DICOM stands for "Digital Imaging and Communications in Medicine". DICOM is the registered

trademark of the National Electrical Manufacturers Association for its standards publications relating to digital communications of medical information.

DICOM Echo – a DICOM message sent from one DICOM entity to another used to test whether DICOM connectivity exists.

DICOMDIR File: A unique and mandatory DICOM File within a File-set which contains the Media Storage Directory SOP Class. This File is given a single component File ID, DICOMDIR.

DICOM File: A DICOM File is a file with a content formatted according to the requirements of DICOM PS 3.10-2004. In particular such files shall contain the File Meta Information and a properly formatted Data Set.

DICOM File Format**:** The DICOM File Format provides a means to encapsulate in a File the Data Set representing a SOP Instance related to a DICOM Information Object.

DICOM File Service: The DICOM File Service specifies a minimum abstract view of files to be provided by the Media Format Layer. Constraining access to the content of files by the Application Entities through such a DICOM File Service boundary ensures Media Format and Physical Media independence.

DIMSE – DICOM Message Service Element. This represents an abstraction of a common set of things that a user would do to a data element, would likely use over and over, and would appear in various different contexts.

DIMSE-C – DICOM Message Service Element—Composite.

DIMSE-C services – A subset of the DIMSE services which supports operations on Composite SOP Instances related to composite Information Object Definitions with peer DIMSE-service-users.

DIMSE-N – DICOM Message Service Element—Normalized.

DIMSE-N services – A subset of the DIMSE services which supports operations and notifications on Normalized SOP Instances related to Normalized Information Object Definitions with peer DIMSE-service-users.

I, J, K, L, M, N, O

Information Object Class **or**

Information Object [Definition] (IOD): A software representation of a real object (e.g., CT Image, Study, etc.). An Information Object is generally a list of characteristics (Attributes) which completely describe the object as far as the software is concerned. The formal description of an Information Object generally includes a description of its purpose and the Attributes it possesses.

Information Object Instance or

Instance (of an IOD): A software representation of a specific occurrence of a real object or entity, including values for the Attributes of the Information Object Class to which the entity belongs.

Little Endian – A term for encoding data where the least-significant byte appears first and remaining bytes follow in ascending order of significance; sometimes known as "Intel" format (see Big Endian).

LUT: Lookup Table.

Message – A data unit of the Message Exchange Protocol exchanged between two cooperating DICOM Application Entities. A Message is composed of a Command Stream followed by an optional Data Stream.

Meta Service-Object Pair (SOP) Class: a pre-defined set of SOP Classes that may be associated under a single SOP for the purpose of negotiating the use of the set with a single item.

Meta SOP Class: A collection or group of related SOP Classes identified by a single Abstract Syntax UID, which, when taken together, represent a logical grouping and which are used together to provide a high-level functionality, e.g., for the purpose of negotiating the use of the set with a single item.

Module: A logical group of the valid attributes of DICOM information objects.

NEMA – National Electrical Manufacturers Association.

Normalized Information Object: A DICOM Information Object (see IOD) whose attributes contain a single real world object. *Note: the differentiation of normalized versus composite information object definitions is not strongly enforced in DICOM 3.0.*

P, Q, R, S

Presentation Context – A Presentation Context consists of an Abstract Syntax plus a list of acceptable Transfer Syntaxes. The Presentation Context defines both what data will be sent (Abstract Syntax) and how the data are encoded to be sent (Transfer Syntax).

Protocol Data Unit (PDU) – A data object which is exchanged by software protocol devices (entities, machines) within a given layer of the protocol stack.

Query – see **C-FIND.**

Real-World Activity: Something which exists in the real world and which pertains to specific area of information processing within the area of interest of the DICOM Standard. A Real-World Activity may be represented by one or more SOP Classes.

Real-World Object: Something which exists in the real world and upon which operations may be performed which are within the area of interest of the DICOM Standard. A Real-World Object may be represented through a SOP Instance.

SCP – DICOM Service Class Provider, often referred to as a "DICOM Server"

SCU – DICOM Service Class User, often referred to as a "DICOM Client"

Service Class: A group of operations that a user might want to perform on particular Information Objects. Formally, a structured description of a service which is supported by cooperating DICOM Application Entities using specific DICOM Commands acting on a specific class of Information Object.

Service Class Provider (SCP, Provider, Server) – A device which provides the services of a DICOM Service Class or Classes which are utilized by another device (SCU) and which performs operations and invokes notifications on a specific Association.

Service Class User (SCU, User, Client) – A device which utilizes the DICOM Service Class or Classes which are provided by another device (SCP) and which invokes operations and performs notifications on a specific Association.

Service-Object Pair (SOP) – The combination of a DICOM Information Object and the Service Class which operates upon that object.

SOP Class – A DICOM term which is identical to an Abstract Syntax; it identifies a set of SOPs which, when taken together, represent a logical grouping (see Meta SOP Class).

Storage Service Class (SSC) – A DICOM term for a logical grouping of Service Classes which all involve storage of images.

T, U, V

Tag – A unique identifier for an element of information composed of an ordered pair of numbers (a Group Number followed by an Element Number), which is used to identify Attributes and corresponding Data Elements.

TCP/IP – Transmission Control Protocol / Internet Protocol.

Transfer Syntax – A part of the DICOM Presentation Context which specifies a set of encoding rules that allow Application Entities to unambiguously negotiate the encoding techniques (e.g., Data Element structure, byte ordering, compression) they are able to support, thereby allowing these Application Entities to communicate.

Unique Identifier (UID) – A globally unique identifier (based on the structure defined by ISO 8824 for OSI Object Identifiers) which is assigned to every DICOM information object as specified by the DICOM Standard (see PS 3.5, Appendices B and C) and which guarantees global unique identification for objects across multiple countries, sites, vendors and equipment.

Value Multiplicity (VM) - The Value Multiplicity of a data element indicates the number of values encoded in the element.

Value Representation (VR) – A VR is the defined format of a particular data element.

W, X, Y, Z

– End of Glossary –

