

Vital 8.1 SP1

Installation and Migration Instructions



Last updated August 9, 2018

Introduction

These are instructions for the installation of Vital 8.1 SP1, and for the migration to Vital 8.1 SP1 from earlier versions of Vital.

You can migrate directly to 8.1 SP1 from version 8.1, but not from earlier versions. If you wish to migrate from a version prior to 8.1, you must first migrate to 8.1, and then migrate to 8.1 SP1.

Migration to 8.1 SP1 from versions prior to 8.1 cannot be carried out using these instructions; instead, the following migration paths and their associated instruction documents must be used:

From 5.4.X: Migrate 5.4.X to 6.1; then migrate 6.1 to 6.2; then migrate 6.2 to 8.1; then migrate 8.1 to 8.1 SP1.

From 5.3: Migrate 5.3 to 5.4; then migrate 5.4 to 6.1; then migrate 6.1 to 6.2; then migrate 6.2 to 8.1; then migrate 8.1 to 8.1 SP1.

From 5.2.X: Migrate 5.2.X to 5.3; then migrate 5.3 to 5.4; then migrate 5.4 to 6.1; then migrate 6.1 to 6.2; then migrate 6.2 to 8.1; then migrate 8.1 to 8.1 SP1.

From 3.4.X: Migrate 3.4.X to 5.4.1; then migrate 5.4.1 to 6.1; then migrate 6.1 to 6.2; then migrate 6.2 to 8.1; then migrate 8.1 to 8.1 SP1.

Note: Vital 8.1 SP1 includes Vitation citation software. Vitation is installed automatically when you install Vital, and no additional configuration is needed for you to use basic Vitation functionality. For Vitation customization instructions please see the [Vital user documentation](#).

Vital 8.1 SP1

Installation and Migration Instructions



Important Notes

Please request your software files from Innovative Customer Support at least one week prior to performing your upgrade. Innovative Interfaces® software Vital uses a strict, quality-driven procedure for software version control and distribution, and we will not be able to honor rush requests for upgrade software. Most software requests take at least three working days to process.

Please read these upgrade instructions completely before performing your upgrade, and contact Innovative Customer Support in advance of your upgrade if you have any questions.

Note the following recent important changes in Vital:

- Starting with version 8.1 SP1, Vital uses HTTP Secure (HTTPS) by default. The Vital 8.1 SP1 distribution includes a self-signed SSL certificate, and the **server.xml** and **web.xml** files are configured to use HTTPS.
- Starting with version 8.1, Vital's use of Docker requires that you use Red Hat Enterprise Linux 7 (RHEL7) or CentOS 7 (CentOS7) as the host operating system. RHEL6 and CentOS6 are NOT supported.
- Also starting with version 8.1, Vital no longer supports the use of MySQL databases. If your Vital installation uses a MySQL database, you must move your data to a PostgreSQL database before migrating to Vital 8.1.
- Also starting with version 8.1, Vital's use of Docker requires that you run a command to access Vital's directory structure within the Docker container. You must run this command in order to access the **vital.properties** file, Vital log files, and other resources associated with your Vital installation. See the section [Modifying and Executing the Run Command](#) for more information.
- Oracle is no longer supported, as of Vital 8.0.
- Authority control is not supported, as of Vital 8.0. If your institution currently uses authority control, you should NOT migrate to Vital 8.1 SP1. Contact Innovative Customer Support for more information.
- You cannot migrate previously customized template files to Vital 8.1 SP1. Vital 8.1 SP1 provides you a graphical user interface that you can use to customize some aspects of the Vital Access Portal interface.
- By default, locale strings from previous versions are not migrated to Vital 8.1 SP1. Contact Innovative Customer Support if you need assistance moving customized locale files to Vital 8.1 SP1.

These instructions are based on the installation of Vital version 8.1 SP1, with notes on migration from previous versions.

Vital 8.1 SP1

Installation and Migration Instructions



Table of Contents

Introduction.....	i
Important Notes	ii
Table of Contents.....	iii
Preliminary Instructions	1
Installing Docker	2
Installing Docker on the Server.....	2
Other Helpful Docker Commands.....	3
Building the Docker Image and Installing Vital.....	4
Migrating to Vital 8.1 SP1 from an Earlier Version	5
Copying Customized XSL Files or indexing.xml to Vital 8.1 SP1.....	8
Starting the Vital Docker Container	10
Modifying and Executing the Run Command	10
Example Scenario.....	14
Moving or Copying Files to Your Vital 8.1 SP1 Installation	16
Moving or Copying Files Via a Temporary Docker Container	16
Post-Installation Instructions	19
Additional Migration Notes	21
Statistics Migration	21
Policies Migration	21
Batch Tool Migration.....	21
Reports Migration.....	21
Template and Configuration Migration	21
Appendix A: Purging the Contents of the Vital Database.....	22
Appendix B: Backing Up and Restoring Docker Volumes	23
Backing Up Data Volumes.....	23
Restoring Backed Up Data Volumes.....	24

Vital 8.1 SP1

Installation and Migration Instructions



Preliminary Instructions

Recommended workflow:

If you are migrating from a previous version, it is best to first perform the migration on a test installation of Vital. This way you can compare functionality and customizations of both versions.

1. If you have not already downloaded the Vital software, please request and download the tar file and license from the Innovative FTP server in binary format. Please make sure you request the license from Innovative, prior to installation. This license is needed for Vital to function.
2. Copy the Vital tar file that you downloaded (in this example, **docker-vital-8.1.SP1-rf6cce50.20180808-linux-x86_64.tar.gz**) into this directory and then untar it:

```
$ /opt/vtls/: tar -xzf docker-vital-8.1.SP1-rf6cce50.20180808-linux-x86_64.tar.gz
```

3. Consult your copy of **installation.xml** from your previous installation if you wish to use the same passwords, ports, etc., as your previous installation.

Vital 8.1 SP1

Installation and Migration Instructions



Installing Docker

In order to install Vital 8.1 SP1, the latest version of Docker must be installed on your server.

In the section [Installing Docker on the Server](#), we provide a workflow for installing Docker on your server.

You can use the commands in the section [Other Helpful Docker Commands](#) to determine whether Docker is already installed on your server, and, if it is present, to determine the Docker version number.

Installing Docker on the Server

To install Docker on your server:

1. Open a command-line terminal and log in to your server.

2. Install required packages:

```
sudo yum install yum-utils device-mapper-persistent-data lvm2
```

3. Set up the **stable** repository:

```
sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

4. Create the **docker** user group:

```
sudo groupadd docker
```

5. Add your current Linux user to the **docker** group:

```
sudo usermod -aG docker [user name]
```

6. Refresh your user's group permissions by closing your terminal session and then logging in to the server again.

7. Update the package cache:

```
sudo yum makecache fast
```

8. Install Docker:

```
sudo yum install docker-ce
```

Vital 8.1 SP1

Installation and Migration Instructions



9. Verify that the version of Docker installed is version 17.12.1-ce or later:

```
docker version
```

10. Run Docker as a service:

```
sudo systemctl start docker
```

11. Verify that the Docker service is running:

```
ps -ef | grep docker
```

Once you have verified that Docker is installed and running, continue to the section [Building the Docker Image and Installing Vital](#).

Other Helpful Docker Commands

Check to see whether Docker has been installed on your server:

```
which docker
```

If Docker is present, check the Docker version number:

```
docker version
```

Once you have verified that the latest version of Docker is installed and running, continue to the section [Building the Docker Image and Installing Vital](#).

Vital 8.1 SP1

Installation and Migration Instructions



Building the Docker Image and Installing Vital

Notes:

- We recommend that the **/var/lib** directory on your server have 20 GB of free space available to handle building the Docker image.
- See the “Building Vital docker image” section of the **readme.md** file for additional information.

To build the Vital 8.1 SP1 Docker image and install Vital 8.1 SP1:

1. Navigate to the location on your server where you extracted the files from the Vital 8.1 SP1 tar file.
2. Enter the **docker** directory:

```
cd docker
```

3. If you want to use a PID namespace other than ‘vital’, edit the **PID_NAMESPACE** value in the **config** file.

Note: This value corresponds to the **<pidNamespace>** setting in the old Vital installation script.

4. If you plan to use handles, place your **admpriv.bin** file in the **docker** directory. The **build.sh** script will copy it to the **/home/dbadmin** directory.
5. Run the **build.sh** script to build the ‘vital’ Docker image and install Vital:

```
./build.sh
```

If you are building the Vital Docker image for the first time, the **build.sh** script will take approximately one hour to run.

Note that by default, the **build.sh** script creates a Vital Docker image with an image name of **vital:8.1.SP1**. You can build a Docker image with a different image name, by using the **--version** flag in your build command:

```
./build.sh --version [image name]
```

6. When the **build.sh** script finishes running,
 - If you are migrating to Vital 8.1 SP1 from an older version of Vital, continue to the section [Migrating to Vital 8.1 SP1 from an Earlier Version](#).
 - If you are installing Vital for the first time, continue to the section [Starting the Vital Docker Container](#).

Vital 8.1 SP1

Installation and Migration Instructions



Migrating to Vital 8.1 SP1 from an Earlier Version

After you have [built the Vital 8.1 SP1 Docker image](#), follow the instructions in this section to migrate to Vital 8.1 SP1 from Vital 8.1. Note that these instructions assume that your Vital 8.1 SP1 Docker image has an image name of **vital:8.1.SP1**.

Reminder: See the instructions in the [Introduction](#) for migrating from versions prior to Vital 8.1. The instructions for migration contained in this document do not apply to migration from versions prior to Vital 8.1.

Note that your existing `vital_batch` and `vital_batch_mod` data volumes are not modified during the migration process. This means that if you have customized files in these volumes in Vital 8.1, your customizations are retained and are present after you upgrade to Vital 8.1 SP1.

To migrate to Vital 8.1 SP1 from Vital 8.1:

1. Open a command-line terminal and log in to your server.
2. (Optional) If you have customized the **indexing.xml** file, XSL stylesheet files, or template files (**Contact.vm**, **Copyright.vm**, **Disclaimer.vm**, and **Privacy.vm**) you can copy your custom files to your new Vital 8.1 SP1 installation. To do this:
 - a. Create a directory called **81files** on your server and then use SFTP to copy the customized files from your Vital 8.1 container to the **81files** directory.
 - b. Run the following command to change the permission settings of the **81files** directory and the files within it:

```
chmod -R a+rx 81files
```

After you migrate to Vital 8.1 SP1, use the instructions in the [Copying Customized XSL Files or indexing.xml to Vital 8.1 SP1](#) section to copy the **indexing.xml** file and/or XSL stylesheets to your new installation. Use the instructions in the [Moving or Copying Files to your Vital 8.1 SP1 Installation](#) section to copy customized template files to the Vital 8.1 SP1 container via SFTP.

3. Stop the Vital 8.1 container:

```
docker stop [container identification number]
```

Note: Instead of typing the entire container identification number in the command above, you can enter the first few characters of the number.

Vital 8.1 SP1

Installation and Migration Instructions



4. Copy the following sample upgrade command and paste it into a text editor for editing:

```
docker run -it --rm \  
  -p 8080:8080 -p 8005:8005 -p 8443:8443 -p 8182:8182 -p 8183:8183 -p 389:389 \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo///g')" \  
  -e "DBADMIN_PASSWORD=dbadmin" \  
  -e "VITAL_HOSTNAME=localhost" \  
  -e "VITAL_PORT=8080" \  
  -e "FEDORA_ADMIN_PASSWORD=fedoraAdmin" \  
  -v dbadmin_home_81sp1:/home/dbadmin \  
  -v dbadmin_home_81:/home/dbadmin_old \  
  -v vital_db_81:/opt/vtls/vital/psql \  
  -v vitation_81:/opt/vtls/vital/applications/vitation_old \  
  --name vital_container_upgrade \  
  vital:8.1.SP1 upgrade
```

5. Lines starting with '-v' indicate data volumes. Edit each of the following lines so that each data volume name matches the name used in your Vital 8.1 installation:

```
-v dbadmin_home_81:/home/dbadmin_old \  
-v vital_db_81:/opt/vtls/vital/psql \  
-v vitation_81:/opt/vtls/vital/applications/vitation_old \  

```

In the example lines above, the suffix '_81' has been appended to each data volume name.

6. (Optional) Edit the following line to assign a volume name that reflects the new Vital 8.1 SP1 installation:

```
-v dbadmin_home_81sp1:/home/dbadmin \  

```

7. If necessary, edit the last line of the command, which specifies the Vital 8.1 SP1 Docker image name:

```
vital:8.1.SP1 upgrade
```

This value must match the image name value that was displayed when the Vital 8.1 SP1 [image build was completed](#). By default, this is 'vital:8.1.SP1'. If you used the `--version` flag in your build command, your image would have received a different name, which should be used as the value in this line.

8. Copy the entire edited upgrade command from your text editor and paste it in at your server prompt.
9. Hit Enter to run the command.

The terminal screen displays output while the command is running. When the command has successfully finished running, the screen displays a success message.

Vital 8.1 SP1

Installation and Migration Instructions



10. If you have customized XSL files or a customized **indexing.xml** file that you would like to copy to your Vital 8.1 SP1 installation, continue to the section [Copying Customized XSL Files or indexing.xml to Vital 8.1 SP1](#). Otherwise, continue to the section [Starting the Vital Docker Container](#).

Vital 8.1 SP1

Installation and Migration Instructions



Copying Customized XSL Files or indexing.xml to Vital 8.1 SP1

If you have customized XSL files or the **indexing.xml** file, you can copy the custom files to your Vital 8.1 SP1 installation. You should copy these files to your 8.1 SP1 installation after [running the upgrade command](#), but before starting Vital 8.1 SP1 for the first time.

Note that you should have already copied your custom files from the Vital 8.1 container into the **81files** directory on your server. See step 2 of the [Migrating to Vital 8.1 SP1 from an Earlier Version](#) section for more information.

To copy customized XSL or **indexing.xml** files:

1. Run a temporary container with a bind mount. To do this, first copy the following sample run command and paste it into a text editor window for editing:

```
docker run -it --rm \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo\\\\\\\\g') " \  
  -e "MOUNT_DIR=/opt/vtls/mounteddata" \  
  --mount \  
type=bind,source=/home/dbadmin/hostcustomervital,target=/opt/vtls/mounteddata \  
  -v dbadmin_home:/home/dbadmin \  
  --entrypoint /bin/bash \  
  --name temp_vital_container \  
  vital
```

2. Edit the `--mount type` line of the command above:

```
--mount \  
type=bind,source=/home/dbadmin/hostcustomervital,target=/opt/vtls/mounteddata \  

```

Replace `'/home/dbadmin/hostcustomervital'` with the path to the **81files** directory on your server where your customized files reside.

3. Edit the `'-v dbadmin_home'` line of the command above:

```
-v dbadmin_home:/home/dbadmin \  

```

Replace `'dbadmin_home'` with the volume name you used in the upgrade command (for example, `'dbadmin_home_81sp1'`).

4. Edit the `'vital'` line of the command:

```
vital
```

Vital 8.1 SP1

Installation and Migration Instructions



This line specifies the image name. Change the 'vital' value to the image name value that was displayed when the [image build was completed](#). By default, this is 'vital:8.1.SP1'. If you used the `--version` flag in your build command, your image would have received a different name, which should be used as the value in this line.

5. Copy **indexing.xml** from the bind mount (host) to the data container located at ``$VITAL_CONFIG_DIR``:

```
cp $MOUNT_DIR/indexing.xml $VITAL_CONFIG_DIR
```

6. Copy stylesheets from the bind mount (host) to the data container located at ``$VITAL_CONFIG_DIR``:

```
cp -R $MOUNT_DIR/customized.xsl $VITAL_CONFIG_DIR/stylesheets
```

7. Exit the temporary container:

```
exit
```

8. Continue to the section [Starting the Vital Docker Container](#).

Vital 8.1 SP1

Installation and Migration Instructions



Starting the Vital Docker Container

Start the Vital Docker container, using the run command provided below. In the workflow below we provide an example of the run command, which you must edit to reflect your Vital installation.

The **readme.md** file, found in the **docker** directory, contains additional information about the run command.

Every line beginning with '-e', as well as the ports (line beginning with '-p') can be changed if necessary each time the container is started.

We discuss each line in the run command in the [Modifying and Executing the Run Command](#) section below. We also provide an [example scenario](#) which describes how you might use a single Docker image to run two Docker containers, one housing a testing Vital instance and one housing a production Vital instance.

Modifying and Executing the Run Command

1. Copy the following sample run command and paste it into a text editor for editing:

```
docker run -d --rm \  
  -p 8080:8080 -p 8005:8005 -p 8443:8443 -p 8182:8182 -p 8183:8183 -p 389:389 \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo\\\\\\\\g') " \  
  -e "DBADMIN_PASSWORD=dbadmin" \  
  -e "VITAL_HOSTNAME=localhost" \  
  -e "VITAL_PORT=8080" \  
  -e "FEDORA_ADMIN_PASSWORD=fedoraAdmin" \  
  -v dbadmin_home:/home/dbadmin \  
  -v vital_objects:/opt/vtls/vital/store \  
  -v vital_db:/opt/vtls/vital/psql \  
  -v vital_batch:/opt/vtls/vital/batch \  
  -v vital_batch_mod:/opt/vtls/vital/batch-modification \  
  -v vitation:/opt/vtls/vital/applications/vitation \  
  -v vital_fedora_data:/opt/vtls/vital/applications/fedora/data \  
  -v vital_solr_data:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
  -v vital_site_assets:/opt/vtls/vital/applications/fedora/server/apache-  
tomcat-7.0.69/webapps/ROOT/sites \  
  --name vital_container \  
  vital
```

2. Edit the port numbers in the port (-p) line of the command, as needed:

```
-p 8080:8080 -p 8005:8005 -p 8443:8443 -p 8182:8182 -p 8183:8183 -p 389:389 \  

```

This line specifies the default external and internal ports.

Vital 8.1 SP1

Installation and Migration Instructions



8080:8080 = Vital port
8005:8005 = shutdown port
8443:8443 = redirect port
8182:8182 = image server port (http)
8183:8183 = image server port (https) (only necessary if running Vital with https)
389:389 = LDAP port (only necessary if using LDAP to log into Vital)

If any of these ports is already in use on your server, you can change the first port in that pair to a different port number. For example, if port 8080 is in use, edit the pair to read "9080:8080". If port 389 is in use, edit the pair to read "9389:389".

Note: The first three port settings (Vital port, shutdown port, redirect port) correspond to Fedora Settings (3. Server Port; 4. Shutdown Port; 5. Redirect Port) in the installation script used in previous versions of Vital.

3. The next line of the run command sets Vital's time to the server's time:

```
-e "TZ=$(readlink /etc/localtime | sed 's/^.*zoneinfo\\///g') " \
```

There is no need to modify this line.

4. Edit the dbadmin password value, as needed:

```
-e "DBADMIN_PASSWORD=dbadmin" \
```

This line specifies the dbadmin password, which corresponds to the dbadmin user that is used within the container. This user is the owner of all of the files within the container.

5. Edit the hostname line of the command, as needed:

```
-e "VITAL_HOSTNAME=localhost" \
```

'localhost' should be changed to the server hostname (for example, 'www.mylibrarysite.com'). The value configured here will be used in various configuration files in Vital.

Note: This line corresponds to the Hostname (External) setting (Global Settings -> 3. Hostname (External)) in the installation script used in previous versions of Vital.

6. Edit the Vital port line of the command, as needed:

```
-e "VITAL_PORT=8080" \
```

Vital 8.1 SP1

Installation and Migration Instructions



The `VITAL_PORT` should match the first port number in the first pair in the port (-p) line of the command, above. If the value in the first line was left at default ('8080:8080'), then this line can be left at the default setting. If, for example, the default was changed to '9080:8080', then the port number in this line must be changed to '9080'.

7. Edit the `fedoraAdmin` password line of the command, as needed:

```
-e "FEDORA_ADMIN_PASSWORD=fedoraAdmin" \
```

This line allows you to configure the `fedoraAdmin` password.

Note: This line corresponds to Fedora Settings -> 2. Administrative Password (`fedoraAdmin`) in the installation script used in previous versions of Vital.

8. Lines starting with '-v' indicate data volumes. These volumes are preserved when the container is stopped.
 - a. If you are installing Vital for the first time and are running only one Docker container, these lines can be left at their defaults.
 - b. If you are running multiple Docker containers (for example, one housing a testing Vital instance and one housing a production Vital instance), you will have to configure each data volume with a suffix to distinguish it from other containers' volumes. In the example below, the suffix '_81sp1test' has been appended to each volume name:

```
-v dbadmin_home_81sp1test:/home/dbadmin \
-v vital_objects_81sp1test:/opt/vtls/vital/store \
-v vital_db_81sp1test:/opt/vtls/vital/psql \
-v vital_batch_81sp1test:/opt/vtls/vital/batch \
-v vital_batch_mod_81sp1test:/opt/vtls/vital/batch-modification \
-v vitation_81sp1test:/opt/vtls/vital/applications/vitation \
-v vital_fedora_data_81sp1test:/opt/vtls/vital/applications/fedora/data \
-v vital_solr_data_81sp1test:/opt/vtls/vital/applications/apache-solr-
4.10.4/data \
-v
vital_site_assets_81sp1test:/opt/vtls/vital/applications/fedora/server/apache-
tomcat-7.0.69/webapps/ROOT/sites \
```

- c. If you are migrating to Vital 8.1 SP1 from Vital 8.1, edit each of the following lines so that each data volume name matches the name used in your Vital 8.1 installation:

```
-v vital_objects_81:/opt/vtls/vital/store \
-v vital_db_81:/opt/vtls/vital/psql \
-v vital_batch_81:/opt/vtls/vital/batch \
-v vital_batch_mod_81:/opt/vtls/vital/batch-modification \
-v vitation_81:/opt/vtls/vital/applications/vitation \
```

Vital 8.1 SP1

Installation and Migration Instructions



```
-v vital_fedora_data_81:/opt/vtls/vital/applications/fedora/data \  
-v vital_solr_data_81:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
-v vital_site_assets_81:/opt/vtls/vital/applications/fedora/server/apache-  
tomcat-7.0.69/webapps/ROOT/sites \  

```

In the example lines above, the suffix '_81' has been appended to each data volume name. In addition, edit the following line so that the 'dbadmin_home' value matches the value specified for 'dbadmin_home' in the [Vital 8.1 SP1 upgrade command](#):

```
-v dbadmin_home_81sp1:/home/dbadmin \  

```

9. Edit the container name line:

```
--name vital_container \  

```

Change 'vital_container' to a unique name (for example, 'vital_81sp1test').

10. Edit the last line of the command, which specifies the image name:

```
vital
```

Change this value to the image name value that was displayed when the Vital 8.1 SP1 [image build was completed](#). By default, this is 'vital:8.1.SP1'. If you used the `--version` flag in your build command, your image would have received a different name, which should be used as the value in this line.

11. Save the edited run command as a file with the filename **runit.sh**.
12. Use SFTP to place the **runit.sh** file in the directory where you extracted the files from the Vital 8.1 SP1 tar file.
13. Navigate to the location on your server where you extracted the files from the Vital 8.1 SP1 tar file.
14. Run the following command to give all other users execute permission:

```
chmod a+x runit.sh
```

15. Run the **runit.sh** script to execute the run the command and start Vital:

```
./runit.sh
```

The system starts Vital. The terminal displays a 64-character alphanumeric string. This is the container identification number for the Docker container that you have just started.

16. (Optional) If you want to access Vital log files or configure **vital.properties** settings for your new Vital instance, run the following command:

Vital 8.1 SP1

Installation and Migration Instructions



```
docker exec -it [container identification number] /bin/bash
```

The `exec` command allows you to access the Vital directory structure inside the Docker container.

Note: Instead of typing the entire container identification number in the command above, you can enter the first few characters of the number.

17. If you are migrating to Vital 8.1 SP1 from an older version of Vital, continue to the section [Moving or Copying Files to Your Vital 8.1 SP1 Installation](#). If you are installing Vital for the first time, continue to the section [Post-Installation Instructions](#).

Example Scenario

You have built an image, 'vital:8.1.SP1', and you wish to use it to run two Docker containers, one housing a testing Vital instance and one housing a production Vital instance.

For the production server, configure the run command to use the default ports, data volumes with suffix '_81splprod', and container name 'vital_81splprod', and then execute the run command:

```
docker run -d --rm \  
  -p 8080:8080 -p 8005:8005 -p 8443:8443 -p 8182:8182 -p 8183:8183 -p 389:389 \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo//g')" \  
  -e "DBADMIN_PASSWORD=dbadmin" \  
  -e "VITAL_HOSTNAME=www.mylibrary.com" \  
  -e "VITAL_PORT=8080" \  
  -e "FEDORA_ADMIN_PASSWORD=fedoraAdmin" \  
  -v dbadmin_home_81splprod:/home/dbadmin \  
  -v vital_objects_81splprod:/opt/vtls/vital/store \  
  -v vital_db_81splprod:/opt/vtls/vital/psql \  
  -v vital_batch_81splprod:/opt/vtls/vital/batch \  
  -v vital_batch_mod_81splprod:/opt/vtls/vital/batch-modification \  
  -v vitation_81splprod:/opt/vtls/vital/applications/vitation \  
  -v vital_fedora_data_81splprod:/opt/vtls/vital/applications/fedora/data \  
  -v vital_solr_data_81splprod:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
 \  
  -v vital_site_assets_81splprod:/opt/vtls/vital/applications/fedora/server/apache- \  
  tomcat-7.0.69/webapps/ROOT/sites \  
  --name vital_81splprod \  
  vital:8.1.SP1
```

Vital 8.1 SP1

Installation and Migration Instructions



For the test server, configure the run command to use non-default ports, data volumes with suffix '_81sp1test', and container name 'vital_81sp1test', and then execute the run command:

```
docker run -d --rm \  
  -p 8280:8080 -p 8205:8005 -p 8243:8443 -p 8282:8182 -p 8283:8183 -p 2389:389 \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo///g')" \  
  -e "DBADMIN_PASSWORD=dbadmin" \  
  -e "VITAL_HOSTNAME=www.mylibrary.com" \  
  -e "VITAL_PORT=8280" \  
  -e "FEDORA_ADMIN_PASSWORD=fedoraAdmin" \  
  -v dbadmin_home_81sp1test:/home/dbadmin \  
  -v vital_objects_81sp1test:/opt/vtls/vital/store \  
  -v vital_db_81sp1test:/opt/vtls/vital/psql \  
  -v vital_batch_81sp1test:/opt/vtls/vital/batch \  
  -v vital_batch_mod_81sp1test:/opt/vtls/vital/batch-modification \  
  -v vitation_81sp1test:/opt/vtls/vital/applications/vitation \  
  -v vital_fedora_data_81sp1test:/opt/vtls/vital/applications/fedora/data \  
  -v vital_solr_data_81sp1test:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
 \  
  -v vital_site_assets_81sp1test:/opt/vtls/vital/applications/fedora/server/apache-  
tomcat-7.0.69/webapps/ROOT/sites \  
  --name vital_81sp1test \  
  vital:8.1.SP1
```

Vital 8.1 SP1

Installation and Migration Instructions



Moving or Copying Files to Your Vital 8.1 SP1 Installation

Note: The workflows described in this section should not be used to migrate objects and datastreams from a previous version. To migrate objects and datastreams from a previous version, see the section [Migrating to Vital 8.1 SP1 from an Earlier Version](#).

We suggest two workflows for moving or copying files to your Vital 8.1 SP1 installation, which is running inside a Docker container:

- [Moving or copying files via a temporary Docker container](#). This workflow is most often used to move or copy configuration files, model files, and data files for use with the Batch Tool. (Note that you may instead use SFTP to copy files for use with the Batch Tool.)
- Moving or copying files via SFTP from within the Docker container. Use this workflow to copy customized template files (**Contact.vm**, **Copyright.vm**, **Disclaimer.vm**, and **Privacy.vm**) to your new installation, to download the Vital license, or for any other situation that requires the transfer of files into the Docker container.

If the files you intend to move or copy are located on the same server as your Vital 8.1 SP1 installation, you can use either workflow.

If the files you intend to move or copy are NOT located on the same server as your Vital 8.1 SP1 installation, transfer objects via SFTP from within the Docker container.

Before you begin either workflow, verify that you have a [Vital Docker container running](#), and identify the directory that contains the files you wish to move to your Vital 8.1 SP1 installation. We will refer to this directory as ``$SOURCE_DIR`` in these instructions.

Moving or Copying Files Via a Temporary Docker Container

Note: Use this workflow only if the files you intend to move or copy are located on the same server as your Vital 8.1 SP1 installation. If your files are located on a different server, use SFTP to move or copy files from within the Docker container.

You will be creating a temporary container to move batch load files from your machine (the host) to the ``vital_batch`` data container. You will then run the Batch Tool inside the actual running container. At the end of this, you will have used the Batch Tool to upload files into Vital.

See the **batch_load.md** file, found in the **docker/docs** directory, for more information.

Vital 8.1 SP1

Installation and Migration Instructions



To move or copy files via a temporary Docker container:

1. Give all read, write, execute permissions for everything in `SOURCE_DIR`

```
[dbadmin@host] chmod -R a+rwX $SOURCE_DIR
```

2. Run a temporary container with a bind mount:

```
[dbadmin@host] docker run -it --rm \
  -e "TZ=$(readlink /etc/localtime | sed 's/^.*/zoneinfo///g')" \
  -e "MOUNT_DIR=/opt/vtls/mounteddata" \
  --mount
type=bind,source=/home/dbadmin/hostcustomerdata,target=/opt/vtls/mounteddata \
  -v vital_batch:/opt/vtls/vital/batch \
  --entrypoint /bin/bash \
  --name temp_vital_container \
  vital
```

Note: In the above example, `/home/dbadmin/hostcustomerdata` is `SOURCE_DIR`.

3. From inside the temporary Docker container, copy or move data from the bind mount (host) to the data container located at `VITAL_BATCH_DIR`.

- a. To copy data, use the following command:

```
[dbadmin@temp_vital_container] cp -R $MOUNT_DIR/batchloadfiles $VITAL_BATCH_DIR
```

- b. To move data, use the following command:

```
[dbadmin@temp_vital_container] mv $MOUNT_DIR/batchloadfiles $VITAL_BATCH_DIR
```

4. Exit the temporary Docker container:

```
[dbadmin@temp_vital_container] exit
```

5. Enter the Docker container that is running Vital 8.1 SP1:

```
[dbadmin@host] docker exec -it vital_container /bin/bash
```

6. Enter `VITAL_BATCH_DIR`:

```
[dbadmin@vital_container] cd $VITAL_BATCH_DIR
```

Vital 8.1 SP1

Installation and Migration Instructions



7. Run the Batch Tool to upload files into Vital 8.1 SP1.

Note: To free up disk space, you may choose to delete files inside the `VITAL_BATCH_DIR` AFTER uploading them into Vital using the Batch Tool. Make sure to verify that the files are in Vital BEFORE deleting files from this directory.

8. Exit the running Vital container:

```
[dbadmin@vital_container] exit
```

Vital 8.1 SP1

Installation and Migration Instructions



Post-Installation Instructions

Note: As of version 7.0, Vital comes with apache-tomcat-7.0.69 as an update to the previously used apache-tomcat-7.0.61. Please keep this in mind when navigating to files in the **opt/vtls/vital/applications/fedora/server/apache-tomcat-7.0.69** path.

1. Apply your Vital license to this installation. Innovative will have provided you with a file entitled **vital.license**. You may reuse your previous license if it is for the same hostname. Otherwise, request a new license from Innovative. To apply your license,

- a. While logged in to your server, run the `exec` command:

```
docker exec -it [container identification number] /bin/bash
```

This command allows you to access the Vital directory structure inside the Docker container (see [Starting the Vital Docker Container](#)).

Note: Instead of typing the entire container identification number in the command above, you can enter the first few characters of the number.

- b. Navigate to the **opt/vtls/vital/config/license** directory.
- c. Retrieve the Vital license and place it in the **opt/vtls/vital/config/license** directory:

```
$ opt/vtls/vital/config/license: ftp ftp.vtls.com
ftp> cd /pub/customers/DIRECTORY_NAME
ftp> get vital.license
ftp> bye
```

2. Please refer to the section [Additional Migration Notes](#) for additional notes on migrating the contents of an 8.1 installation into this new 8.1 SP1 installation.
3. Load the Vital URL in a web browser to verify that the Vital instance is running.

`http://www.mylibrarysite.com:8080/vital/access/manager/Index`

In the sample Vital URL above, 'www.mylibrarysite.com' represents the `VITAL_HOSTNAME` specified in the hostname line of the run command. '8080' represents the `VITAL_PORT` specified in the Vital port line of the run command.

Vital 8.1 SP1

Installation and Migration Instructions



You will not see your objects immediately, but they will appear when they are automatically committed to Vital's index.

4. To use Valet with Vital 8.1 SP1, you must perform the following steps:
 - a. Modify the Valet **publish.xml** file to include the hostname and port where the Vital 8.1 SP1 Docker container is running. This specifies the location to which Valet submits material. See the Valet user documentation for more information.
 - b. Remove the **deny-apim-if-not-localhost.xml** policy file from the **/opt/vtls/vital/store/fedora-xacml-policies/repository-policies/default** directory.
 - c. Run the **fedora.sh** script to reload policies:

```
./fedora.sh policies
```
5. If you have configured Vital to use handles (persistent identifiers for digital objects), you should review the handles settings in the **vital.properties** and **vital.properties.site** files after migrating to a new version. Note that if you have changed the server port during the migration, you must either:
 - Update the port number in the **vital.handle.format** setting found in the **vital.properties.site** configuration file (if using handles for the first time and migrated data contains no handles).
 - OR-
 - Add a new entry to **vital.handle.format** that matches the port of existing handles (if migrated data already contains handles).

If you have any questions, please contact Customer Support.

Vital 8.1 SP1

Installation and Migration Instructions



Additional Migration Notes

For migrating customizations, it is best to compare the old files with the new and move or update individual snippets. Always remember to back up the original files included with your new Vital installation before making changes and read the **README.txt** file located in your **/opt/vtIs/vital/migration/docs** directory for additional instructions.

Statistics Migration

In 8.1 SP1, statistics are migrated by the migration script.

Policies Migration

In Vital 8.1 SP1, policies are migrated by the migration script.

Batch Tool Migration

In Vital 8.1 SP1, batch load configuration files are migrated by the migration script.

Reports Migration

In 8.1 SP1, reports are migrated by the migration script.

Template and Configuration Migration

If you have customized the following template files, you can migrate the customized files to Vital 8.1 SP1: **Contact.vm**, **Copyright.vm**, **Disclaimer.vm**, and **Privacy.vm**. See the [Migrating to Vital 8.1 SP1 from an Earlier Version](#) section for more information.

You cannot migrate other previously customized template files to Vital 8.1 SP1. However, Vital 8.1 SP1 provides you a graphical user interface that you can use to customize some aspects of the Vital Access Portal interface.



Appendix A: Purging the Contents of the Vital Database

Follow the instructions below if you wish to purge the Vital database of all content related to sites, users, and groups, and recreate the tables with a properly-constructed yet empty Vital database.

Note that we recommend that you back up all of your existing content before proceeding.

1. Edit the **applications/flyway-4.0/conf/flyway.conf** file to comment out the **flyway.cleanDisabled** line:

```
# Whether to disabled clean. (default: false)

# This is especially useful for production environments where running clean can
# be quite a career limiting move.

# flyway.cleanDisabled=true

# The version to tag an existing schema with when executing baseline. (default:
# 1)
```

2. Clear all database tables by running the following command:

```
applications/flyway-4.0/flyway clean
```

3. Recreate the database tables by running the following command:

```
applications/flyway-4.0/flyway migrate
```

Vital 8.1 SP1

Installation and Migration Instructions



Appendix B: Backing Up and Restoring Docker Volumes

You can back up Docker data volumes and later restore the data volumes from your back-ups.

Backing Up Data Volumes

To back up Docker data volumes:

1. Open a command-line terminal and log in to your server.
2. Stop the Vital Docker container:

```
docker stop [container identification number]
```

Note: Instead of typing the entire container identification number in the command above, you can enter the first few characters of the number.

3. Copy the following sample `backup` command and paste it into a text editor for editing:

```
docker run -it --rm \  
  -e "TZ=$(readlink /etc/localtime | sed 's/^\.*zoneinfo\\///g')" \  
  -e "MOUNT_DIR=/opt/vtls/mountedata" \  
  --mount type=bind,source=/home/dbadmin/backups,target=/opt/vtls/mountedata \  
  -v dbadmin_home:/opt/vtls/source \  
  -v vital_objects:/opt/vtls/vital/store \  
  -v vital_db:/opt/vtls/vital/psql \  
  -v vital_batch:/opt/vtls/vital/batch \  
  -v vital_batch_mod:/opt/vtls/vital/batch-modification \  
  -v vitiation:/opt/vtls/vital/applications/vitiation \  
  -v vital_fedora_data:/opt/vtls/vital/applications/fedora/data \  
  -v vital_solr_data:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
  -v vital_site_assets:/opt/vtls/vital/applications/fedora/server/apache-tomcat-7.0.69/webapps/ROOT/sites \  
  --name vital_container \  
  vital backup
```

Vital 8.1 SP1

Installation and Migration Instructions



4. Edit the `--mount type` line of the command to change the `/home/dbadmin/backups` path so that it reflects the location where you want to store backup files.

Note: The path you specify must have file permissions set to '777'.

5. Copy the entire edited `backup` command from your text editor and paste it in at your server prompt.
6. Hit Enter to run the command.

The terminal screen displays output while the command is running. When the command has successfully finished running, the screen displays a success message.

Restoring Backed Up Data Volumes

To restore backed up Docker data volumes:

1. Verify that the location on your server where your backups are stored contains the following files:

```
dbadmin_home.tar.gz
vital_db.tar.gz
vital_site_assets.tar.gz
vital_batch.tar.gz
vital_fedora_data.tar.gz
vital_solr_data.tar.gz
vital_batch_mod.tar.gz
vital_objects.tar.gz
vitation.tar.gz
```

2. Copy the following sample restore command and paste it into a text editor for editing:

```
docker run -it --rm \
  -e "TZ=$(readlink /etc/localtime | sed 's/^\.*zoneinfo\\///g')" \
  -e "MOUNT_DIR=/opt/vtls/mounteddata" \
  --mount type=bind,source=/home/dbadmin/backups,target=/opt/vtls/mounteddata \
  -v dbadmin_home:/opt/vtls/source \
  -v vital_objects:/opt/vtls/vital/store \
  -v vital_db:/opt/vtls/vital/psql \
  -v vital_batch:/opt/vtls/vital/batch \
  -v vital_batch_mod:/opt/vtls/vital/batch-modification \
  -v vitation:/opt/vtls/vital/applications/vitation \
  -v vital_fedora_data:/opt/vtls/vital/applications/fedora/data \
  -v vital_solr_data:/opt/vtls/vital/applications/apache-solr-4.10.4/data \
```

Vital 8.1 SP1

Installation and Migration Instructions



```
-v vital_site_assets:/opt/vtls/vital/applications/fedora/server/apache-tomcat-7.0.69/webapps/ROOT/sites \  
--name vital_container \  
vital restore
```

3. Edit the `--mount type` line of the command to change the `/home/dbadmin/backups` path so that it reflects the location where your backup files are stored.
4. Lines starting with `-v` indicate data volumes. To prevent overwriting data, edit each of the following lines to assign a unique name to each data volume (`dbadmin_home`, `vital_objects`, etc.):

```
-v dbadmin_home:/opt/vtls/source \  
-v vital_objects:/opt/vtls/vital/store \  
-v vital_db:/opt/vtls/vital/psql \  
-v vital_batch:/opt/vtls/vital/batch \  
-v vital_batch_mod:/opt/vtls/vital/batch-modification \  
-v vitation:/opt/vtls/vital/applications/vitation \  
-v vital_fedora_data:/opt/vtls/vital/applications/fedora/data \  
-v vital_solr_data:/opt/vtls/vital/applications/apache-solr-4.10.4/data \  
-v vital_site_assets:/opt/vtls/vital/applications/fedora/server/apache-tomcat-7.0.69/webapps/ROOT/sites \  

```

5. Copy the entire edited `backup` command from your text editor and paste it in at your server prompt.
6. Hit Enter to run the command.

The terminal screen displays output while the command is running. When the command has successfully finished running, the screen displays a success message.