

Siebel

Deploying Siebel CRM Containers

November 2023



Siebel
Deploying Siebel CRM Containers

November 2023

F87476-03

Copyright © 1994, 2023, Oracle and/or its affiliates.

Author: Sukanya Mishra

Contents

Get Help i

1 What's New in This Release 1

What's New in Deploying Siebel CRM Containers, Siebel CRM 23.11 Update	1
What's New in Deploying Siebel CRM Containers, Siebel CRM 23.7 Update	1
What's New in Deploying Siebel CRM Containers, Siebel CRM 22.7 Update	1
What's New in Deploying Siebel CRM Containers, Siebel CRM 22.6 Update	2

2 Deploying Siebel CRM Containers 3

Deploying Siebel CRM Containers	3
About Deploying Siebel CRM Containers	3
Performing Prerequisite Tasks	4
Downloading the Siebel Container Images from ARU	7
Setting Up the Siebel Containers in Docker	7
Configuring and Running the Siebel Applications	9
Stopping the Siebel Containers	12
Upgrading Containers to the Latest Version of Siebel CRM	12
Updating the Siebel Database Using the PostInstallDBSetup Utility	13
Applying POC Patches	17
Migrating Persistent Volume Content	19

Get Help

Preface

This preface introduces information sources that can help you use the application and this guide.

Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at <https://docs.oracle.com/>.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle Accessibility Program website](#).

Contacting Oracle

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit [My Oracle Support](#) or visit [Accessible Oracle Support](#) if you are hearing impaired.

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to:
oracle_fusion_applications_help_ww_grp@oracle.com.

1 What's New in This Release

What's New in Deploying Siebel CRM Containers, Siebel CRM 23.11 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
<i>Using the PostInstallDBSetup-19.list File</i>	Modified topic. Describes how to configure the PostInstallDBSetup process for non-ENU languages.

What's New in Deploying Siebel CRM Containers, Siebel CRM 23.7 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
<i>Applying POC Patches</i>	New topic. Describes the steps to follow to patch your containers with respect to the POC content supplied to you.

What's New in Deploying Siebel CRM Containers, Siebel CRM 22.7 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
<i>Details of Running PostInstallDBSetup</i>	Modified topics. Corrected a command element in the docker command.

What's New in Deploying Siebel CRM Containers, Siebel CRM 22.6 Update

The following information lists the changes in this revision of the documentation to support this release of the software.

Topic	Description
<i>About Deploying Siebel CRM Containers</i> Multiple topics	New and modified topics. This document pertains to Siebel CRM 22.6 Update and later, and has been updated to take into account the switch to Oracle Database 19c. The publication method for this book has changed and the book is now included in the <i>Siebel Bookshelf</i> .

2 Deploying Siebel CRM Containers

Deploying Siebel CRM Containers

This document describes the steps required to download Oracle's Siebel CRM container images from ARU (Automated Release Update system) and to deploy and run these containers on a new virtual machine (VM). It includes the following topics:

- *About Deploying Siebel CRM Containers*
- *Performing Prerequisite Tasks*
- *Downloading the Siebel Container Images from ARU*
- *Setting Up the Siebel Containers in Docker*
- *Configuring and Running the Siebel Applications*
- *Stopping the Siebel Containers*
- *Upgrading Containers to the Latest Version of Siebel CRM*
- *Updating the Siebel Database Using the PostInstallDBSetup Utility*
- *Migrating Persistent Volume Content*

About Deploying Siebel CRM Containers

Note: This document pertains to Siebel CRM 22.6 and later, to take into account the switch to Oracle Database 19c.

Note: Before proceeding, customers obtaining and preparing to deploy Siebel CRM using the methods described here are strongly encouraged to read this entire document and to familiarize themselves with the various tools that are used in the deployment process. Oracle does not certify Docker or other container management systems for the uses described here. The capability described in this document is available as limited availability (LA) functionality, available only to qualified customers who are installing Siebel CRM as described here. This document was first published for Siebel CRM 19.9, with the status of controlled availability (CA).

This document describes the steps required to download Oracle's Siebel CRM container images from ARU (Automated Release Update system) and to deploy and run these containers on a new virtual machine (VM). Your deployment might be on-premise, cloud-based, or some combination of these. Although Docker is mentioned, you can run the Siebel containers using Docker, Kubernetes, Podman, or another container management system. An ARU request is required for accessing ARU from My Oracle Support (MOS).

This document provides a worked example for deploying Siebel CRM 22.x in a new deployment. In this example, you first set up a virtual machine using Oracle VM VirtualBox, install Oracle Linux, and prepare the system further. Then you download the Siebel CRM container images from ARU, load these containers into Docker, set up the Oracle Database, and configure and run the Siebel applications.

Your specific deployment choices might vary. Although there are similarities to deploying Docker images to OCI or other clouds or to doing bare metal installs, exploring those options is beyond the scope of this document. If you get

stuck, then it is recommended that you first follow this guide as written to gain familiarity with all that is involved before moving to your target architecture.

Note: Deploying Siebel CRM using containers takes the place of standard installation tasks for a new deployment. For information about other tasks that you perform, see *Siebel Installation Guide*, the *Siebel CRM Update Guide and Release Notes* on My Oracle Support for your release, and other applicable documents on *Siebel Bookshelf* or My Oracle Support. See also documentation for Oracle Linux, Oracle Database, Oracle VM VirtualBox, and third-party products such as Docker or Git.

Perform the following process steps to get the containers up and running (Steps 1 through 4), to stop the containers (Step 5), and to upgrade the containers (Steps 6 through 8):

1. *Performing Prerequisite Tasks*
2. *Downloading the Siebel Container Images from ARU*
3. *Setting Up the Siebel Containers in Docker*
4. *Configuring and Running the Siebel Applications*
5. *Stopping the Siebel Containers*
6. *Upgrading Containers to the Latest Version of Siebel CRM*
7. *Updating the Siebel Database Using the PostInstallDBSetup Utility*
8. *Migrating Persistent Volume Content*

Related Books

Siebel CRM Update Guide and Release Notes on My Oracle Support

Siebel Installation Guide

Siebel Database Upgrade Guide

Performing Prerequisite Tasks

Perform the tasks in this topic to prepare an Oracle Linux virtual machine for further steps in setting up the Siebel containers that you obtain from ARU.

Installing Oracle Linux Server on a Virtual Machine

This task describes how to install Oracle Linux Server on a virtual machine. You can skip this task if you already have Oracle Linux installed and running on a virtual machine.

1. Create a virtual machine in a suitable environment. Use Oracle VM VirtualBox or another suitable virtualization system. These instructions assume the use of VirtualBox.
2. Install Oracle Linux as a guest operating system for this virtual machine, using the latest update of Oracle Linux 7 Server software that you obtain from Oracle Software Delivery Cloud:

`https://edelivery.oracle.com`

3. Set a password for the root user.
4. Create a new administrative user, such as *demoadmin* (example).
5. Run the `visudo` command to provide superuser privileges (sudo access) to *demoadmin*.
6. Restart the virtual machine and log in as *demoadmin*.

7. (Optional) Take a snapshot of the virtual machine, which creates an entry of a step, to be able to get back to that step in the future.
 - a. Click Take to open the Take a Snapshot of Virtual Machine dialog box.
 - b. Give a name to the snapshot, such as Oracle Linux Installation and Upgrade.

Note: Later procedures also mention taking a snapshot at a particular point in the process. These are optional steps that give you a backup at each point, so that you can revert to a previous snapshot if you make a mistake.

Perform the remaining tasks in this section as `demoadmin` to further prepare the system for deploying the Siebel CRM containers.

Enabling Permissive Mode for SELinux

Security-Enhanced Linux (SELinux) is a Linux kernel module that supports access-control security policies. To facilitate your deployment process, it is recommended to change from *enforcing* to *permissive* mode, which means that you are warned about issues but that actions are not blocked. Consult your security team for specific requirements that might apply for your company. This is an optional step.

Enabling the ol7_addons Yum Repository

You must enable the Yum repository `ol7_addons`, which gives you access to install Docker.

1. First, use the following command to edit the file `public-yum-ol7.repo`:

```
sudo vi /etc/yum.repos.d/public-yum-ol7.repo
```

2. Within the `[ol7_addons]` section of the file `public-yum-ol7.repo`, set the `Enable` parameter to a value of 1, as shown in this example section of the file:

```
[ol7_addons]
name=Oracle Linux $releasever Add ons ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL7/addons/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1
```

Installing the Docker Engine

Use the following command to install the Docker Engine using Yum:

```
sudo yum install -y docker-engine
```

Starting the Docker Service

Use the following commands to start the Docker service automatically and to restart it automatically when the virtual machine starts:

```
sudo systemctl start docker
sudo systemctl enable docker
```

Enabling the Administrative User

Enable the administrative user `demoadmin` to execute Docker commands. To do this, use the following command:

```
sudo usermod -aG docker demoadmin
```

You will need to log out and log in again before the permission is active.

Installing Git

Install the Git tool to allow pulling software from GitHub. To do this, use the following command:

```
sudo yum install -y git
```

Testing the Docker Engine and Git

Perform the following steps to test access to the Docker Engine and whether it is up and running.

1. Restart the virtual machine.
2. Log in to the Oracle Linux virtual machine as the demoadmin user.
3. Run the following command:

```
docker --version
```

For example, the response might be as follows:

```
Docker version 19.03.11-ol, build 9bb540d
```

4. Run the following command:

```
git --version
```

For example, the response might be as follows:

```
git version 1.8.3.1
```

5. (Optional) Take a snapshot: Prerequisites Completed.

Enabling VirtualBox Additions

Enable the VirtualBox additions on the virtual machine to improve the integration of the host (local computer) and the client (hosted virtual machine). Do the following to enable the VirtualBox additions:

1. Install kernel additions to enable the virtual machine to recognize the host's screen resolution. Use the following command:

```
sudo yum install kernel-uek-devel-$(uname -r) -y
```

2. Insert the VirtualBox Guest Additions CD using the menu option Devices > Insert Guest Additions CD Image.
3. Map the network drive using the menu option Devices > Shared Folders > Shared Folders Settings. Mapping the directory on the host allows you to easily move any content from the host to the virtual machine.
4. Add the demoadmin user to the VirtualBox shared folders group so it has read and write access to the files in the shared folders. Use the following command:

```
sudo usermod -aG vboxsf demoadmin
```

Log off the client (virtual machine) and log in again.

5. Enable the transfer of container images between the host and the client (virtual machine). To do this, use the menu option Devices > Shared Clipboard > Bidirectional.

Later, you will copy content from the host (local computer where you are downloading the container images) to the client (virtual machine) where you will run the containers.

6. (Optional) Take a snapshot: VirtualBox additions installed.

Downloading the Siebel Container Images from ARU

This topic describes the steps to download the Siebel container images (also called Siebel containers) from ARU (Automated Release Update system) and prepare to import them into Docker.

1. Drill down on the ARU patch number.
2. Review the applicable instructions and then click Download to download the applicable files from ARU. The filenames differ for each Siebel CRM monthly release. The filenames use the following format:
 - `ptrackingID_2200_Linux-x86-64_1of2.zip`
 - `ptrackingID_2200_Linux-x86-64_2of2.zip`

Below are example filenames for Siebel CRM 22.6. The filenames might differ for your Siebel CRM 22.x release.

- `p34269720_2200_Linux-x86-64_1of2.zip`
 - `p34269720_2200_Linux-x86-64_2of2.zip`
3. Unzip these files. For Siebel CRM 22.6, for example, the following files are created:
 - `siebel-22.6.tar.gz`
 - `deploy-db-19c-22.6.tar.gz`
 4. (Optional) Take a snapshot: Containers Downloaded from ARU.

Setting Up the Siebel Containers in Docker

This topic describes the steps to load the Siebel containers into Docker. This topic contains the following information:

- *Setting Up Oracle Database*
- *Importing the Siebel Containers into the Docker Engine*
- *Creating Directories for Persistence Volumes that Store the Container Configurations*

Setting Up Oracle Database

This topic is part of *Setting Up the Siebel Containers in Docker*.

In this task, the content from the database image is unpacked to a directory accessible to Oracle Database, after which the Oracle Database is instructed to plug in that supplied database. Oracle Database refers to any viable Oracle Database instance that can plug in the supplied database. The Oracle Database instance could be running in a container or could represent a standard installed database instance.

To set up Oracle Database, do the following:

1. Log in to the following:

```
https://container-registry.oracle.com
```

2. Select Oracle Database Enterprise Edition.
3. Accept the license agreement.
4. Enter the login command and supply your single sign-on (SSO) credentials:

```
docker login container-registry.oracle.com
```

5. Enter the `pull` command. The command for Oracle Database 19c is:

```
docker pull container-registry.oracle.com/database/enterprise:19.3.0.0
```

This completes the Oracle Database setup.

Importing the Siebel Containers into the Docker Engine

This topic is part of *Setting Up the Siebel Containers in Docker*.

In this task, you import the Siebel containers into the Docker Engine.

1. Run the following commands to import the Siebel container images after you download them from ARU. For Siebel CRM 22.6, for example, load the following files:

```
docker load < siebel-22.6.tar.gz
docker load < deploy-db-19c-22.6.tar.gz
```

These commands unpack the files downloaded from ARU and create containers in Docker that you can use to create the Siebel Enterprise. After you have done this, the following containers are now available:

- **Siebel Engine Container.** This container is created from the file that starts with *siebel*. This file represents the Siebel Enterprise Components. This container can be started in one of three modes: CGW (for Siebel Gateway), SES (for Siebel Server), or SAI (for Siebel Application Interface).
 - **Siebel Sample Database.** This container is created from the file that starts with *deploy-db*. This file represents a pluggable database for Oracle Database 19c.
2. To validate the result of this step, run the following command:

```
docker images
```

Sample output from this command might resemble the following:

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
store/oracle/siebel	22.6	4cf7d46dbc00	2 weeks ago	3.4GB
store/oracle/siebel/deploy-db/19c/sample	22.6	4b60f409a7e8	2 weeks ago	2.15GB

3. (Optional) Take a snapshot: Containers Imported into Docker.

Creating Directories for Persistence Volumes that Store the Container Configurations

This topic is part of *Setting Up the Siebel Containers in Docker*.

To be able to persist the configuration of the Siebel Enterprise Server and Oracle Database, you must create directories that will serve as *persistence volumes*. A persistence volume is, effectively, the saved state of the container. These are the files that reside outside of the Docker images and store one possible configuration of that image when instantiated as a container.

1. Decide where to create the persistence volume directories.
2. Use the `mkdir` command to create separate directories to store the configuration data. Create the following directories:
 - **ORCL19**. Include this directory if you are using the containerized database. You can omit this directory if you are using an existing database: for example, DBaaS or ATP.
 - **ENT**. Create this directory to contain all of the subdirectories pertinent to a single Siebel Enterprise. Under this directory, create the following directories:
 - **CGW**. Siebel Gateway (sometimes called the Cloud Gateway).
 - **SAI**. Siebel Application Interface.
 - **SES**. Siebel Server.
 - **SFS**. Siebel File System.

Storing the enterprise configuration in the persistence volumes means that you do not need to reconfigure the enterprise after the virtual machine or the containers restart. They will load the persistent state and carry on from where they left off. You can also transport the enterprise state to other servers by copying the persistent volumes to another server.

Configuring and Running the Siebel Applications

This topic describes tasks for configuring and running the Siebel applications. It contains the following information:

- [Downloading Scripts from GitHub](#)
- [Deploying and Launching the Siebel Containers](#)
- [Loading the Sample Siebel Database](#)
- [Configuring Siebel CRM](#)

Downloading Scripts from GitHub

This topic is part of [Configuring and Running the Siebel Applications](#).

To download example scripts from GitHub, do the following:

1. Create a directory in which to store scripts that will facilitate running and configuring the Siebel applications, for example, `siebel`. Enter a command like the following:

```
mkdir ~/siebel
```
2. Change directory to `siebel`, as follows:

```
cd ~/siebel
```
3. Copy and clone the `ConfiguringSiebel.git` repository into the directory you created, as follows:

```
git clone https://github.com/OracleSiebel/ConfiguringSiebel.git
```

4. Change directory as follows:

```
cd ~/siebel/ConfiguringSiebel/Containerization/Docker/oraclelinux/classic-mde/manage/  
scripts/automate-architecture/launch-siebel
```

This directory contains the configuration file `sample-architecture-19.sh` and scripts for launching the Siebel applications. This set of scripts encapsulates all the necessary commands to launch the Siebel containers.

Deploying and Launching the Siebel Containers

This topic is part of *Configuring and Running the Siebel Applications*.

Parameters to launch and configure the environment are stored in a configuration file, such as `sample-architecture-19.sh`. In almost all cases, you will need to read and edit the parameters in this file. To do this, it is recommended to first copy this file to a new name, as appropriate for your deployment.

To set the environment definitions in the configuration file:

1. As appropriate, copy the `sample-architecture-19.sh` file to a new name (for example, `siebert.sh`), and make appropriate changes to the parameters in the file copy. The main parameters you are likely to need to change are the following:

- **PV.** Set PV to point to the location you chose to store persistent volumes. For example:

```
PV=/var/lib/docker/volumes/PV
```

- **ROOT.** Set ROOT to point to the location where the `automate-architecture` folder exists, which is part of the launch system you cloned from GitHub. For example:

```
ROOT=/home/demoadmin/dev/oraclesiebel/oraclelinux/manage/scripts/automate-architecture
```

- **DOMAIN.** Set DOMAIN to match the domain for which the SSL certificate was generated. For containers obtained from ARU, set this to `company.com`. If you are building your own containers, then adjust appropriately. The value `siebel.local` is correct only if you build your container using the supplied build system from GitHub.

```
DOMAIN=siebel.local
```

- **CONTAINERBASE.** Set CONTAINERBASE to match the image name, which you can see using the `docker images` command. For containers obtained from ARU, set this to `store/oracle/siebel`.

```
CONTAINERBASE=registry.local.com:5000/siebel
```

- **COMPGRPS.** Set COMPGRPS to the component groups that will be configured using the `configure` command later in this guide.

```
COMPGRPS=callcenter,siebelwebtools,eai
```

2. Once the environment parameters are set, start the containers for the enterprise using a command like the following:

```
bash startAll19 version parameter_file
```


For example, you might use the following command for Siebel CRM 22.6:

```
bash startAll19 22.6 siebent.sh
```

The example script `startAll19` launches four containers. First, it launches the database engine, then the SAI container, the CGW container, and finally the SES container.

The given setup assumes that the `sample-architecture-19.sh` parameter file (or `siebent.sh` in the example) describes the base location for the persistent volumes using the variable `PV`. The `PV` directory contains a directory named by the `ENTERPRISE` parameter, and this directory contains the subdirectories `CGW`, `SES`, `SAI`, and `SFS`. Create these directories before the first run, as described in *Creating Directories for Persistence Volumes that Store the Container Configurations*. These directories must be writeable by the user who will run the scripts, as defined by the parameter `RUNASUSER`.

If the persistence directories are empty on the first run, then they are populated during the launch of the containers. If they were populated by a previous run, then the contents already present are used. This is the primary mechanism by which the state is saved for the enterprise.

Loading the Sample Siebel Database

This topic is part of *Configuring and Running the Siebel Applications*.

At this stage, you have a running database and a set of running Siebel containers. Now you must provide the Siebel database to connect to, such as the sample database provided.

1. Change directory as follows:

```
cd ~/siebel/ConfiguringSiebel/Containerization/Docker/oraclelinux/classic-mde/manage/  
scripts/automate-architecture/db
```

2. Inspect the content of the script `dropDB19`. Make sure that the `dbFileLocation` parameter, which specifies the persistence volume directory location for the database, is set correctly. For example:

```
dbFileLocation=/var/lib/docker/volumes/persistence/ORCL19/ORCLCDB
```

3. Double-check the exact name of the `deploy-db` container by using the following command:

```
docker images
```

4. Drop the database into location, adjusting for the name of the container you see in your list. Use a command like the following, where `22.x` stands for the actual version number, such as `22.6`:

```
bash dropDB19 store/oracle/siebel/deploy-db/sample:22.x
```

The progress of this operation is shown in several lines of output. At the end of the process, you are returned to the UNIX command line.

Configuring Siebel CRM

This topic is part of *Configuring and Running the Siebel Applications*.

After the containers complete the launch process, before you can configure them, they must have a healthy state, as reported by the following command:

```
docker ps
```

At this point, you can configure the enterprise manually using Siebel Management Console (SMC). SMC is available at the IP address of the machine that launched the enterprise containers at the port specified in the PORT parameter in sample-architecture-19.sh or your copy of it. Enter a URL resembling the following:

```
https://machine-IP:port/siebel/smc
```

To avoid manual setup through SMC, a configure script is also provided that automates setup using the scripts in the `smc` directory. To execute this, use a command like the following:

```
bash configure sample-architecture-19.sh
```

Stopping the Siebel Containers

To stop the Siebel containers for a specific setup of Siebel CRM, do the following:

1. Stop the containers for the installed version of Siebel CRM, using a command like the following:

```
bash stopAll parameter_file
```

For example:

```
bash stopAll sample-architecture-19.sh
```

2. (Optional) Take a snapshot: Siebel Fully Installed on Docker.

Upgrading Containers to the Latest Version of Siebel CRM

For an existing deployment, to upgrade your Siebel containers to the latest version of Siebel CRM 22.x, do the following:

1. Stop the containers for the installed version of Siebel CRM, using a command like the following:

```
bash stopAll sample-architecture-19.sh
```

2. Take a copy of your persistent directory content to back up the state of your enterprise. For example:

```
tar -czvf ENT.tar /var/lib/docker/volumes/PV/ENT
```

3. Upgrade to the latest version (for example, Siebel CRM 22.6) using a command like the following, where 22.x stands for the actual version number, such as 22.6:

```
bash startAll19 22.x sample-architecture-19.sh
```

See also the remaining topics in this document. Read carefully all sections in this document and take any appropriate action before starting containers on the next version.

For complete instructions for any future release, see the updated version of this document.

Updating the Siebel Database Using the PostInstallDBSetup Utility

Follow the process described in this topic to execute, as part of the update process, the required step of running PostInstallDBSetup for Siebel CRM deployments using containers. This topic contains the following information:

- [Overview of Running PostInstallDBSetup](#)
- [Details of Running PostInstallDBSetup](#)
- [Using the PostInstallDBSetup-19.list File](#)
- [Example Output for Running PostInstallDBSetup](#)

Overview of Running PostInstallDBSetup

This topic is part of [Updating the Siebel Database Using the PostInstallDBSetup Utility](#).

PostInstallDBSetup is a tool which serves to apply schema changes required by a given Siebel CRM update to a specific database. Run the PostInstallDBSetup utility for each monthly Siebel CRM 22.x release that you install. Follow the process described in this topic to execute this crucial step in the update process for Siebel CRM deployments using containers.

Note: The standard Siebel CRM installer automates this process for on-premise deployments: the installer for Siebel CRM 22.x releases automatically executes PostInstallDBSetup for each Siebel Enterprise Server installation. You must execute PostInstallDBSetup manually when using containers, as described here, or implement some level of automation yourself.

The container image for Siebel CRM holds the necessary binaries and data to effect the relevant schema changes. Because this process is in the form of *run-once-to-completion*, it is not integrated with other launch systems or container initialization, but is treated as a separate process for the administrator to use. You can automate running PostInstallDBSetup, where required, because it is safe to run this utility even though new schema updates are not delivered in every monthly release.

Before you begin, review the assumptions about your environment for the purposes of this document. It is assumed that you are already using containers to connect to the database to which you need to apply the schema changes, and that you already have a persistent volume which has important files that will be required by PostInstallDBSetup. If you do not already have a configured Siebel Enterprise using containers, then establish one before proceeding.

Process of Running PostInstallDBSetup

The process of executing PostInstallDBSetup for a container deployment includes the following general steps:

- Attaching the persistent volume to the container
- Preparing and populating a template PostInstallDBSetup.ini file
- Configuring database connectivity
- Executing the PostInstallDBSetup process

- Reviewing generated logs in the connected persistent volume

Details of Running PostInstallDBSetup

This topic is part of *Updating the Siebel Database Using the PostInstallDBSetup Utility*.

This topic describes the command for running PostInstallDBSetup to apply any necessary schema changes. Use a command similar to the following, where 22.x stands for the actual version number, such as 22.6. All commands must be entered on one line.

```
docker run --rm -u 1000 --network siebelnet --env-file PostInstallDBSetup-19.list -v /var/lib/docker/volumes/PV/ENT/SES:/persistent --entrypoint /bin/bash store/oracle/siebel:22.x /config/PostInstallDBSetup.sh
```

Command Elements for Running PostInstallDBSetup

The following table provides information about each of the elements in this command.

Note: If you are using Kubernetes, then convert the described parameters into relevant YAML equivalents, along with the content described in *Using the PostInstallDBSetup-19.list File*.

Command Element	Description
<code>docker run</code>	The command to run a Docker container.
<code>--rm</code>	Removes the container instance when processing is complete.
<code>-u 1000</code>	Specifies the user who runs this command, such as the user with user ID 1000, who must have access to the connected persistent volume.
<code>--network siebelnet</code>	Required only if your database is also running in a Docker container on the same machine. If the database runs on another machine, then you do not need this parameter. Allows this container execution to share a network with the database containers. This example uses Oracle Database 19c as a container running on the siebelnet network. This is required for database connectivity to apply the schema changes.
<code>--env-file PostInstallDBSetup-19.list</code>	Specifies a file supplying numerous parameters required for PostInstallDBSetup, as described in <i>Using the PostInstallDBSetup-19.list File</i> .
<code>-v /var/lib/docker/volumes/PV/ENT/SES:/persistent</code>	The location of the persistent volume for a previously configured SES instance, along with the mount point to attach it to inside the container (<code>/persistent</code>).
<code>--entrypoint /bin/bash</code>	Creates a Bash shell using an alternate command to execute in the container than the one configured by default in the container (which is <code>initSES</code>). This is the shell in which <code>PostInstallDBSetup.sh</code> runs.
<code>store/oracle/siebel:22.x</code>	The name of the container to use, where 22.x stands for the actual version number, such as 22.6.
<code>/config/PostInstallDBSetup.sh</code>	The script that the Bash shell executes.

Command Element	Description

Using the PostInstallDBSetup-19.list File

This topic is part of *Updating the Siebel Database Using the PostInstallDBSetup Utility*.

You must create the PostInstallDBSetup-19.list file in the directory from which you generally launch your containers. The following is an example PostInstallDBSetup-19.list file to get you started.

```
DBHOST=oracle19c
DBPORT=1521
DBINST=SAMPLE
ODBC_DSN=SIEBELDB
TBLO=SIEBEL
TBLOUSER=SIEBEL
SIEBUSER=SADMIN
SSE_ROLE=SSE_ROLE
INDSPC=SIEBEL
TBLSPC=SIEBEL
TBLOPASS=Welcome1
SIEBELUSERPASS=Welcome1
sesServerName=SES
PRIMARY_LANG_CD=ENU
OTHER_LANG_CD=
```

Note: Preinstalled containers from Oracle include an Oracle Database instant client, and the specific parameters here pertain only to running PostInstallDBSetup connected to an Oracle Database. If you are creating your own containers, then you must work through the whole process and make necessary changes in your custom layer where you install any alternate database clients.

Note: Release 23.10 added support for running the PostInstallDBSetup process for non-ENU languages. Specify the primary language configured in your database in PRIMARY_LANG_CD. Additional languages should be specified in OTHER_LANG_CD, comma-separated.

These parameters might be self-explanatory for some administrators tasked with executing PostInstallDBSetup. If necessary, then review the information about database updates in *Siebel Installation Guide* and in *Siebel Database Upgrade Guide*.

Example Output for Running PostInstallDBSetup

This topic is part of *Updating the Siebel Database Using the PostInstallDBSetup Utility*.

The examples that follow show the type of output you can expect from running PostInstallDBSetup:

- Example 1: Running PostInstallDBSetup
- Example 2: PostInstallDBSetup Has Already Run

Example 1: Running PostInstallDBSetup

This example uses a freshly installed sample database for Siebel CRM 22.4 and the PostInstallDBSetup process from the Siebel CRM 22.6 image. All commands must be entered on one line.

```
docker run --rm -u 1000 --network siebelnet --envfile PostInstallDBSetup-19.list -v /scratch/persistent/PV/ENT/SES:/persistent --entrypoint /bin/bash store/oracle/siebel:22.6 /config/PostInstallDBSetup.sh
```

```
Write access confirmed for folder --> /persistent
```

The next two lines returned are errors that repeat for each language you are not using and can be safely ignored. These two lines are followed by information about the execution status.

```
mv: cannot stat '/siebel/mde/siebsrvr/bin/csy/omdefs_sia.run': No such file or directory
ln: failed to create symbolic link '/siebel/mde/siebsrvr/bin/csy/omdefs_sia.run': No such file or directory

[SIEBELDB]
Stage 1 of 3 : Running WSRanking
WSRanking script has already been run on this environment
Stage 2 of 3 : Running SeedSchemaManifest Upgrade
Stage 3 of 3 : Running Workflow DR Upgrade and Migration
*****SUMMARY*****
Execution of WSRanking is Not applicable.
SeedSchemaManifest Execution Completed Successfully.
Workflow DR Upgrade and Migration completed successfully
*****
PostInstallDBSetup Execution HTML Report Location : /siebel/ses/siebsrvr/log/
PostInstallDBSetup_2022-05-19_12-04-05.html
real 3m50.899s
user 2m41.013s
sys 0m45.513s
```

This example uses the PostInstallDBSetup-19.list set of environment variables shown in [Using the PostInstallDBSetup-19.list File](#). The process first confirms access to the persistent volume. It then makes sure that the `siebsrvr/sys/.odbc.ini` file has an appropriate configuration for talking to the database by creating a driver and reference if necessary. In this example, those references were already present, as you can see in the two output lines starting with `SIEBELDB` and `[SIEBELDB]`, as they are located with the `grep` command. After this check, `PostInstallDBSetup.ini` is populated with the contents supplied from `PostInstallDBSetup-19.list` and execution starts.

At the end of the execution, a reference is given to the summary log file (in HTML format), which you can review for more details on the process. Because the `siebsrvr/log` file is persisted, you can find this log file in the persistent directory referenced by your command.

In this example, the command ran with `-v /scratch/persistent/PV/ENT/SES:/persistent`, and so the actual log content is located in a log file similar to `/scratch/persistent/PV/ENT/SES/siebsrvr/PostInstallDBSetup_2022-05-19_12-04-05.html`.

Finally, the process provides an indication of how long the process took: in this case, just under four minutes.

Example 2: PostInstallDBSetup Has Already Run

The output in this second example is what you can expect when running the process a second time and with all the same properties (that is, running it unnecessarily). This time, the system quickly determines that execution is unnecessary and quits. All commands must be entered on one line.

```
docker run --rm -u 1000 --network siebelnet --envfile PostInstallDBSetup-19.list -v /scratch/persistent/PV/ENT/SES:/persistent --entrypoint /bin/bash store/oracle/siebel:22.6 /config/PostInstallDBSetup.sh
```

```
[SIEBELDB]
'PostInstallDBSetup' database final configuration is not required on this instance as
it has already been executed in a prior install.
```

```
real 0m3.109s
user 0m1.907s
sys 0m0.468s
```

Applying POC Patches

Follow the process described in this topic to patch your containers with respect to the POC content supplied to you.

This topic contains the following information:

- [Overview of Applying a POC](#)
- [Details of Applying a POC](#)
- [Discover Which POCs Are Applied](#)
- [Details of POC Log Files](#)

Overview of Applying a POC

This topic is part of [Applying POC Patches](#).

A POC is a patch supplied to you by Oracle Support in order to fix an issue specific to your use of a specific Siebel application version.

Over the course of your use of a specific version of Siebel, you may be supplied with one or more POC patches. One or more patches can be applied at a time. You may run the POC patch process more than once over the course of your use of a specific version of Siebel.

Each POC patch contains one or more files which will add to, or overwrite, files in a Siebel installation.

Details of Applying a POC

This topic is part of [Applying POC Patches](#).

This topic describes how to apply single or multiple POCs to a specific version of a Siebel container.

Scripts for applying one or more POCs are supplied within the container from 23.7 onward. If you wish to apply a POC to earlier versions of Siebel, download the 23.7 container in order to obtain the script, then inspect and adapt, or directly use that script, with the earlier version of Siebel.

To extract the scripts for applying a POC, execute the following commands, where X.Y refers to the Siebel container version for example 23.7, for Siebel GA containers 23.7 or greater, and <local_folder> is the name of the source folder of your choice:

```
id=`docker create store/oracle/siebel:X.Y`
docker cp $id:/config/POCAppl <local_folder>
docker rm $id
```

You should now have a folder containing the following files:

applyPOC

Dockerfile

processPatches

Into this folder, now copy the POC patches you wish to apply. These are delivered via myOracleSupport with file names in the form:

```
P12345678_1200_Linux-x86-64_1of2.zip
```

To apply the POC content, now execute the following command:

```
bash applyPOC -s <source_image> -t  
               <target_image_tag>
```

For example:

```
bash applyPOC -s store/oracle/siebel:23.6 -t  
store/oracle/siebel:23.6-patchesToJune23
```

If, over time, you need to apply more POC patches, this can either be done by adding the POC patches to your existing folder and re-running the above process, or you can create a new folder and apply future patches on top of your patched container. For example:

```
bash applyPOC -s store/oracle/siebel:23.6-patchesToJune23 -t  
store/oracle/siebel:23.6-patchesToOctober23
```

Patches will always be applied in correct order, by year first, then in numerical sequence.

To use the patched container instead of the original container, simply change the command you use to launch Siebel. For example:

```
bash startAll119 23.6-patchedToJune23 sample-architecture-19.sh
```

Discover Which POCs Are Applied

This topic is part of [Applying POC Patches](#).

To review which POCs have been applied, start a patched container and examine the contents of the folder `/config/ProcessedPOC`.

For example:

```
docker run --rm -it --entrypoint /bin/bash store/oracle/Siebel:23.6-patchesToOctober23  
cd /config/ProcessedPOC  
ls
```

Details of POC Log Files

This topic is part of [Applying POC Patches](#).

To review which POCs have been applied, start a patched container and examine the contents of the folder `/siebel/mde/poc_backup`.

For example:

```
docker run --rm -it --entrypoint /bin/bash store/oracle/Siebel:23.6-patchesToOctober23  
cd /siebel/mde/poc_backup
```



```
ls
```

Here you will find a list of directories, one for each POC applied. Within each folder you can review the `copy.log` folder to see a log of the work done by the POC installer.

Migrating Persistent Volume Content

This topic describes tasks you might need to perform for migrating persistent volume content if you are migrating from a prior release of Siebel CRM deployed in Docker containers. It contains the following information:

- *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.2 or Later*
- *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.4 or Later*
- *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 22.6 or Later*

Note: Perform all of the tasks that apply in your deployment, in the sequence presented.

Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.2 or Later

This topic is part of *Migrating Persistent Volume Content*.

The new Siebel CRM installer provided as of Siebel CRM 21.2 optimizes the Siebel CRM release and allows all aspects of Siebel CRM to be installed into a single container, thereby simplifying deployment. This also means that the container has two instances of Apache Tomcat installed: one for SES and CGW operation modes, and one for SAI operation mode.

Prior to Siebel CRM 21.2, the container had a directory called `applicationcontainer`, which is replaced by `applicationcontainer_internal` (for SES and CGW modes) and `applicationcontainer_external` (for SAI mode).

Thus, if you want to use persistent volumes that you have previously configured for pre-21.2 containers with 21.2 or later containers, then you need to rename the `applicationcontainer` directory in your `ses` and `cgw` persistent volume directories to `applicationcontainer_internal`, and to rename the `applicationcontainer` directory in the `sa` persistent volume directory to `applicationcontainer_external`.

Also, the old `server.xml` file configured for SAI for pre-21.2 containers must be replaced, because static Siebel content has moved from `siebel.war` (from which it was unpacked to the `webapps/siebel` directory) to `siebelwebroot`. It is recommended that you delete the `conf` directories for all tiers in order to reset them fully.

To make the necessary changes, create a script similar to the following:

```
mv SAI/applicationcontainer SAI/applicationcontainer_external
mv SAI/applicationcontainer_external/conf SAI/applicationcontainer_external/conf_backup
mv SES/applicationcontainer SES/applicationcontainer_internal
mv SES/applicationcontainer_internal/conf SES/applicationcontainer_internal/conf_backup
mv CGW/applicationcontainer CGW/applicationcontainer_internal
mv CGW/applicationcontainer_internal/conf CGW/applicationcontainer_internal/conf_backup
```

Alternatively, you can empty your persistent volume directories and set up your container deployment again from scratch.

You must make these changes before launching the containers for the first time for Siebel CRM 21.2 or later.

Note: Also perform the steps described in *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.4 or Later* and *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 22.6 or Later*.

Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.4 or Later

This topic is part of *Migrating Persistent Volume Content*.

Note: If you are coming from a version earlier than Siebel CRM 21.2 to Siebel CRM 21.4 or later, then first read and act upon the information in *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.2 or Later*.

Siebel CRM 21.4 and later releases provide a new version of Apache ZooKeeper, which is used for the Siebel Gateway registry. As result, there is a change to the persistent volume content for the container acting as the CGW node. For more information about ZooKeeper and Siebel Gateway, see also *Siebel Installation Guide* on the *Siebel Bookshelf*.

As of Siebel CRM 21.4, the ZooKeeper installation has been placed one level deeper in the directory structure, inside a new directory called `registry`. This `registry` directory contains a directory called `conf`, which holds all state-related content for the ZooKeeper instance, and a directory called `zookeeper`, which holds the ZooKeeper version for this update. In addition, one of the files previously holding registry configuration, called `zoo1.cfg`, is renamed to `registry.cfg`. This `registry.cfg` file holds the location of the `conf` directory, so your existing file must be altered to make this change. You must also add two new properties to the CFG file for this ZooKeeper version.

To make the necessary changes, create a script similar to the following, adjusting the directory locations as required for your deployment:

```
# specify location of CGW persistent volume
PVLocation=/var/lib/docker/volumes/PV/ENT/CGW
# specify current owner of files in CGW folder
CGWOwner=demoadmin
ConfLocation=${PVLocation}/gtwysrvr/registry/conf
# make new conf folder
mkdir -p ${ConfLocation}
# migrate content to new conf folder
cp -r ${PVLocation}/gtwysrvr/zookeeper/version-2 ${ConfLocation}
cp ${PVLocation}/gtwysrvr/zookeeper/myid ${ConfLocation}
cp ${PVLocation}/gtwysrvr/zookeeper/conf/log4j.properties ${ConfLocation}
cp ${PVLocation}/gtwysrvr/zookeeper/conf/zoo1.cfg ${ConfLocation}/registry.cfg
# make changes to registry.cfg
sed -i "s/dataDir=.*dataDir=\/siebel\/mde\/gtwysrvr\/registry\/conf/" >> ${ConfLocation}/registry.cfg
echo "admin.enableServer=false" >> ${ConfLocation}/registry.cfg
echo "multiAddress.reachabilityCheckEnabled=false" >> ${ConfLocation}/registry.cfg
# make ownership changes
chown -R ${CGWOwner}:${CGWOwner} ${PVLocation}/gtwysrvr/registry
```

After copying this script to disk, make appropriate changes at the top of the script before execution, to specify the location on disk of your CGW instance's persistent volume and to specify the current owner of files in your CGW directory. You must run the script using a user with sufficient privileges to execute the commands, which create a directory, copy content, edit files, and finally change file ownership of all files touched by the operation.

Note: Also perform the steps described in *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 22.6 or Later*.

Migrating Persistent Volume Content from Prior Releases to Siebel CRM 22.6 or Later

This topic is part of *Migrating Persistent Volume Content*.

Note: If you are coming from a version earlier than Siebel CRM 21.2 to Siebel CRM 22.6 or later, then first read and act upon the information in *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.2 or Later* and *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.4 or Later*. If you are coming from Siebel CRM 21.2 or 21.3 to Siebel CRM 22.6 or later, then first read and act upon the information in *Migrating Persistent Volume Content from Prior Releases to Siebel CRM 21.4 or Later* before performing this task.

Siebel CRM 22.6 and later releases provide a new version of Apache ZooKeeper, which is used for the Siebel Gateway registry. As result, there is a change to the persistent volume content for the container acting as the CGW node. For more information about ZooKeeper and Siebel Gateway, see also *Siebel Installation Guide* on the *Siebel Bookshelf*.

You only need to take action if you have a need to make custom additions to the `registry.properties` file that ZooKeeper uses. In previous versions, this file was considered persistent, meaning that changes made would persist between Siebel updates. However, because new content is required for Siebel CRM 22.6 or later to function, this model has been changed. Instead, the `registry.properties` file is recreated every time the Siebel container starts.

To customize the `registry.properties` file, you need to take one of two actions:

- **Add a custom container layer.** Using a Docker file, replace the `registry.properties` file stored in the location `/siebel/mde/gtwysrvr/registry/conf`.
- **Provide a `registry.properties.process` file.** The goal of the `registry.properties.process` file is to create the `registry.properties` file. At a minimum, this file must copy `/siebel/mde/gtwysrvr/registry/zookeeper/conf/registry.properties` (consider this a template). However, the `registry.properties.process` file might also make additional changes using standard UNIX tools such as `sed` and `awk` in order to modify the contents.

If you create a file called `registry.properties.process` in the persistence volume for CGW, located in `/scratch/persistent/PV/ENT2/CGW/gtwysrvr/registry/conf`, then this file will be processed before `registry.properties` is read. If `registry.properties.process` fails to create `registry.properties` in the correct location, then the template file will be used, ensuring a working enterprise.

